Otto-von-Guericke University Magdeburg

Faculty of Computer Science
Department of Simulation and Graphics

Diploma Thesis

Towards a Better VIEW: Visual Information Exploration on the Web

Marian Dörk

Supervisors:

Sheelagh Carpendale
Carey Williamson

Department of Computer Science
University of Calgary

Marcel Götze

Faculty of Computer Science
Otto-von-Guericke-University Magdeburg

May 19, 2008

Marian Dörk
Towards a Better VIEW: Visual Information Exploration on the Web
Diploma Thesis, Otto-von-Guericke University Magdeburg, May 2008.

Downloads, demos, and additional documentation:
`http://www.mariandoerk.de/diplomathesis/`

Abstract

Given common search interfaces, it is difficult to gain orientation within large information spaces and explore information along conceptual dimensions such as time, location, and topics. In this thesis, I develop the idea of *visual information exploration on the Web* (VIEW) that aims to support a more active way of finding and exploring information by the means of coordinated visualizations. Drawing from both visual information seeking and exploratory search, I present the design of interactive visualization widgets called *VisGets*. VisGets provide the information seeker with visual overviews of multiple aspects of Web resources and a way to pose a search query visually within the Web-browser. This facilitates the construction of combined queries with temporal, spatial, and semantic constraints. A prototype VIEW system was implemented featuring three linked VisGets—time slider, geographic map, and tag cloud—that are used to visually explore news items aggregated from RSS feeds. Preliminary evaluations have shown positive reactions from information seekers and revealed possible applications for VisGets. Directions for future research on visual information exploration are outlined.

Zusammenfassung

Mit gewöhnlichen Suchschnittstellen ist es schwierig, sich in großen Informationsräumen zu orientieren und Informationen entlang konzeptioneller Dimensionen, wie zum Beispiel Zeit, Ort und Thema, zu erkunden. In dieser Diplomarbeit entwickle ich die Idee der *visuellen Informationsexploration im Web* (VIEW), die mit Hilfe von koordinierten Visualisierungen eine aktivere Form der Informationssuche und -exploration darzustellen versucht. Aufbauend auf vorherige Arbeiten zur visuellen und exploratorischen Informationssuche erstelle ich das Konzept von Visualsisierungs-Widgets oder *VisGets*. VisGets geben der/den Informationssuchenden visuelle Übersichten mehrfacher Aspekte von Webressourcen und eine Möglichkeit, eine Suchanfrage visuell im Web-Browser zu formulieren. Dies erlaubt die Zusammensetzung von kombinierten Anfragen aus zeitlichen, räumlichen und semantischen Bedingungen. Ein prototypisches VIEW-System wurde implementiert, das drei verknüpfte VisGets—Zeitleiste, geographische Karte und Tag-Wolke—aufweist, die dazu verwendet werden, Einträge aus aggregierten RSS-Feeds visuell zu explorieren. Erste Evaluierungen haben positive Reaktionen unter Informationssuchenden und mögliche Anwendungen für VisGets gezeigt. Richtungen für weitere Untersuchungen zur visuellen Informationsexploration werden skizziert.

Acknowledgements

The past months and the time of my studies in general would not have been possible nor as enjoyable without the help, support, and friendship of many people.

First, I would like to thank my supervisors in Calgary, Sheelagh Carpendale and Carey Williamson, for giving me trust, ideas, and advice throughout the thesis project and for giving me the opportunity to work with so many wonderful people at the University of Calgary.

I am also grateful to my supervisor in Magdeburg, Marcel Götze, for the helpful comments and suggestions while I was writing this thesis.

Many thanks go to the people from the Interactions Lab, where I enjoyed lunch table conversations as much as brainstorming sessions. In particular I would like to thank Chris Collins, Mark Hancock, Petra Isenberg, Matthew Tobiasz, and Amy Voida for their valuable help and advice.

Special thanks to Alex, Bubi, Daniela, Joe, Kirsten, Kristin, Marco, Matthias, Michael, Nicole, Olaf, Uwe, Valentin, and all the others for being great friends and partners in crime.

Many thanks to Alexander Bubeck, Christopher Collins, Michael K. W. Hentze, and Uta Hinrichs for reading rough drafts and providing constructive comments.

I would like to thank my parents, sister, and brother for always being supportive of my studies regardless whether abroad or at home.

Thank you, Uta, for encouraging, motivating, and inspiring me, but foremost for being a wonderful person in my life.

Contents

1 Introduction

Searching for information on the World Wide Web is a fundamental task undertaken daily by millions of people around the world. This computer use is expanding into more and more aspects of society, as people increasingly turn to the Web as an immediate source for news, research, and entertainment. While accessing Web-based information has evolved into an almost universal way of getting information, the mechanisms for finding information on the Web are typically bound to text-based search. In this thesis I address the limitations of conventional information seeking on the Web and how interactive visualization can improve this process.

1.1 Motivation

While the seemingly-endless information space of the Web contains diverse rich-media content, finding information on the Web is generally done using an ordinary text query on a search engine. This approach is demonstrably useful, in that people routinely find *something* useful, even if it is not exactly what they were seeking. However, searching can be frustrating when queries return thousands of results, many of which are extraneous. Frustrating searches are even more likely when a person's information need is only vaguely defined. Choosing the right keywords for the search query may be difficult, and the text-based result list itself provides little contextual overview to promote general understanding. A common tactic is to issue multiple (slightly different) queries, and to look mainly at the first few items in the list of results. While this can lead to success, it just as often leads to long and laborious searches with imperfect results.

Today's Web features thriving online communities, rich media content, maturing Semantic Web standards, and an increasing number of geographically referenced resources. This structure is quite different from the early days of the Web, which was dominated by unstructured textual information. Consequently, information published on today's Web is becoming ever more complex: it includes not only multimedia, but rich links between information fragments indicating semantic, social, and spatial relationships. However, the traditional Web search process does not reflect these advances, since search is still primarily textual. The information seeker is confronted with a large, potentially relevant information space; however, it is difficult or impossible for the searcher to gain an overview and orientation, or even understanding using the present search mechanisms.

While humans would be considerably overwhelmed having to process such large volumes of abstract data, information visualizations can make large amounts of data more accessible through visual representations and interaction [21]. Leveraging the predisposition of human cognition towards visual perception, information visualizations convey visually summarized views on large amounts of information that can be more easily and quickly grasped by the information seeker [45]. The rationale for this approach comes in part from visual information seeking [4], where visualization techniques have been shown to enhance query formulation and exploration of databases. The concept is also foreshadowed by the recent inclusion of geographic maps and semantic tag clouds in some Web-based searches.

1.2 Problem Statement and Research Questions

In this thesis I address the shortcomings of present information seeking approaches for the World Wide Web. In particular, I criticize information retrieval for being system-centered and information visualization for not sufficiently embracing the Web as an information space and software platform.

Web-based information retrieval is mostly bound to search and browse tasks, which are typically low-level, laborious, and thus not adequate for an information space as large and diverse as the World Wide Web. Current search systems expect information seekers to make possibly vague information needs explicit in textual search queries and assess large amounts of individual search results. What the information space is composed of is hidden from the searcher and only exposed by means of individual searches. To develop an orientation within the information space, the information seeker engages in a 'hit and miss'-tactic that often impedes him/her in developing an understanding of the available information.

Information systems are more focused on ranking schemes, index creation, and retrieval mechanisms than on the searcher, who is expected to engage in rather tedious tasks. Information seeking is mostly conceptualized around technical constraints rather than on the needs and problems of information seekers. Interestingly, the related field of information visualization addresses the human as the central component of interaction with (visual) information. For some of the limitations of Web-based information retrieval, visualization may provide a way to bridge the gap between a person's needs and system constraints.

While information visualization research has addressed the Web as a possible data source before, constraining assumptions about the structure and the scope of the data have limited the impact of visualization on Web-based information seeking. The challenge of visualizing a distributed and diverse information space, such as the Web, has not been sufficiently solved yet. Furthermore, the Web has not been fully embraced as both a software platform and a data source for visualization.

From these considerations I derive the following research questions:

1. How can visualizations facilitate information seeking on the Web?
2. How can a Web-based visual information exploration system be realized?
3. How would such a system be accepted and evaluated by information seekers?

To find answers to these questions, I have surveyed previous research on information seeking and visualization, then I have developed a concept for information seeking that is supported by interactive visualizations, based on this concept I have designed and implemented a system, which has been subjected to exploratory evaluation.

1.3 Thesis Overview

The remainder of this thesis is structured as follows.

Chapter 2 provides the academic context for the thesis, defining the terminology used for information seeking, and discussing related work from the literature.

Chapter 3 develops the concept of visual information exploration on the Web (VIEW) as an information seeking facilitated by InfoVis widgets (VisGets). Derived from the VIEW concept, design goals for VisGets are outlined.

Chapter 4 presents the appearance and functioning of three initial VisGets, describing the choice of information dimensions, and explaining interaction mechanisms for visual information exploration.

Chapter 5 discusses the implementation of a VIEW system, with particular attention given to the Web-based architecture and the aggregation of Web resources.

Chapter 6 presents the results from an initial user study and an informal focus group that were conducted to assess the potential usefulness of VisGets and to generate new ideas about the VIEW approach.

Finally, Chapter 7 summarizes the contributions of this thesis, and highlights promising areas for future work.

2 Related Work

This chapter provides an overview of the research forming the basis for the work presented in this thesis. In Section 2.1 an introduction to the fields of computer science contributing to information seeking research is presented, followed by a review of four different information seeking approaches (Section 2.2). Next, techniques used in information visualization are examined, along with systems for information seeking (Section 2.3). This chapter concludes with a discussion of previous research into Web-based information visualization (Section 2.4).

2.1 Fundamentals

Information seeking is the use of any kind of system or technique to search for and browse through information. Before discussing different theories of information seeking behavior and possibly supporting visualization systems, this section briefly explains how the study of information seeking draws from several mutually-overlapping research areas within computer science: information retrieval (IR), World Wide Web (WWW), human-computer interaction (HCI), and information visualization (InfoVis)—as shown in Figure 2.1.

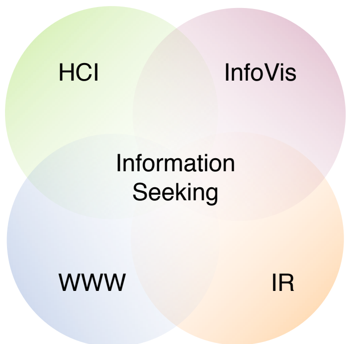


Figure 2.1: Information seeking approached from four computer science disciplines.

2.1.1 Information Retrieval

Historically, information seeking has been the key concern of library science in terms of the organization and provision of printed information sources. With the advent and advancement of computer systems, information seeking is increasingly framed around the more technical concept of information retrieval (IR).

The goal of information retrieval is to retrieve the most relevant documents from a collection according to a query that represents the searcher's information need (see Figure 2.2). The IR system would preprocess and index the document collection and parse the query so that those documents that match the explicitly formulated information need can be retrieved [99]. The quality of the retrieved documents (and hence also the retrieval system) can then be measured with precision and recall: *precision* is a measure for exactness describing the ratio of relevant documents over all retrieved documents and *recall* is a measure of completeness denoting the relation between retrieved documents that are relevant and all relevant documents in the system [106].

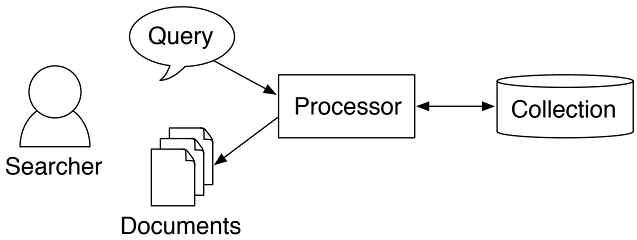


Figure 2.2: Simplified information retrieval process.

Within information retrieval, information seeking has been grouped into two basic user tasks depending on the specificity of the information need: retrieval and browsing [10]. During *retrieval*, the information seeker translates his/her rather specific information need into a search query; a set of documents such as Web pages matching this query (and hopefully the information need) is returned as a ranked list. In *browsing*, the person seeking information does not specify his/her broader information need, which is rather implicitly manifested in the choice of navigation paths. Based on these key concepts and tasks, information retrieval has been especially concerned with the representation of search queries and many documents, and the matching process between them. For example, software architectures behind Web search engines are largely based on information retrieval concepts.

2.1.2 World Wide Web

The appearance of the World Wide Web constitutes a major paradigm shift for information seeking. Before the Web, information systems were mostly centralized databases that had to be accessed through special query languages. The Web, however, became the first interlinked information system that could be accessed from any computer via the Internet. Conceived by Berners-Lee [15], the WWW builds on top of the ideas of associative trails [19] and hypertext [74]. From anywhere in the world, the information seeker can access Web resources that can be stored and provided by Web servers located at any other point in the world—provided both Web server and browser are connected to the Internet (see Figure 2.3). Navigation from one Web page to another is transparent for the information seeker, regardless of whether the pages are stored on the same server or not. There is no complex user interaction required, other than the selection of hyperlinks, to jump within and between Web pages.

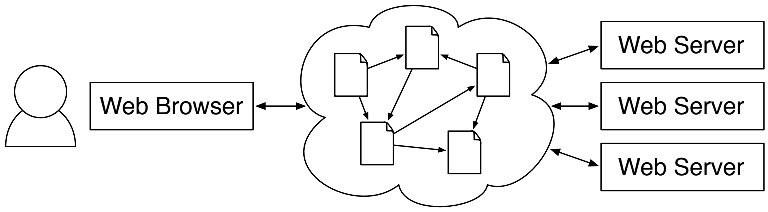


Figure 2.3: World Wide Web: a decentralized information space.

The three essential standards that make the Web work are a markup language for documents, a way to identify and locate distributed resources, and a transfer protocol for exchanging documents:

- The *HyperText Markup Language* (HTML) defines the syntax for creating interlinked documents on the Web [82]. The presentation style of an HTML file can be defined through Cascading Style Sheets (CSS).
- A *Uniform Resource Locator* (URL) represents a Web resource through a human-readable address. This address specifies the protocol, the host, document path, and anchor.
- The *HyperText Transfer Protocol* (HTTP) specifies the procedure for client-server communication on the WWW. The Web browser requests a resource by specifying a URL, and the Web server provides the document in response. The actual transport of the packets containing the data is undertaken by TCP/IP (Transmission Control Protocol and Internet Protocol), as specified in the Internet protocol suite.

New possibilities for information seeking emerge, as Web resources become increasingly structured and the Web itself evolves into a platform. These developments have been called Web 2.0 and Semantic Web, which are described in the following.

Web 2.0

Multiple developments concerning the emerging platform character of the Web have been rather ambiguously summarized under the term *Web 2.0* [78]. In this, the Web is seen in particular as a community and software platform. On one hand, the Web becomes a social platform and a new kind of public space, where the emphasis on information of the Web's early days is accompanied with a notion of community, conversation, and participation. Many people collaborate on projects such as Wikipedia [105], flock to social networking sites like Facebook [30], and publish their own blogs and read those of many others. On the other hand, the Web is regarded as a platform for software development. As the support for Web standards improves and many popular Web sites provide APIs for their data and functionality, Web-based applications can draw from resources that their offline counterparts cannot benefit from.

Semantic Web

The *Semantic Web* has been conceptualized as a long-term process to turn meaning implicitly present on the Web into a more explicit and computer-readable representation [16]. Web-based information already becomes increasingly structured, since Web pages can be marked up with standardized semantic annotations and structured data can be published utilizing maturing Semantic Web formats. Semantic annotations such as Microformats [70] and RDFa [2] can be embedded easily into conventional HTML markup without creating notable overhead for the content provider. It has been suggested that publishing and reusing Semantic Web data may be facilitated by the means of Web 2.0 ideas such as sharing-based communities, mash-ups between different services, and rich browser-side interaction [8]. For example, more powerful filtering and exploration mechanisms can be realized within the browser without requiring a database [50]. In addition to publishing small lightweight information pieces, structured information from large-scale information spaces such as the Wikipedia is extracted and turned into semantic information repositories, which can be queried like a database [9]. Furthermore, the multitude of RSS (Really Simple Syndication) feeds of millions of Web pages contain structured metadata on newly added posts or recent changes on blogs and news sites. These feeds constitute a large interlinked information space with rich and relatively consistent semantics.

Besides the evolving nature of the Web, its wide adoption as an essential information source by people, regardless of age, gender, or origin, is fundamentally changing information seeking. In this light, the WWW draws attention from researchers beyond computer science, particularly from the humanities and social sciences that seek to study human and social behavior in relation to the Web.

2.1.3 Human-computer Interaction

For the study of computer-based information seeking, the field of human-computer interaction (HCI) provides a large range of theories and tools. According to SIGCHI, an ACM special interest group, the HCI community is "concerned with the design, evaluation, and implementation of interactive computing systems for human use and with the study of major phenomena surrounding them" [1]. HCI research topics center around the human using a computer system. Particularly interesting in regard to information seeking are the study of human information processing, interaction design, and evaluation techniques.

Usability and Usefulness

The overall objective of HCI research and practice is to improve the way people interact with computer systems. To achieve this, interactive systems have to become more usable and useful. *Usability* refers to the ease of use, user-friendliness, and accessibility of a system, and *usefulness* denotes whether a system adequately supports the tasks for which the system was intended. However, both criteria are interdependent, since a system is hardly useful, if it is not usable [61]. Particularly HCI research, however, includes user studies and evaluations that assess almost solely the usability of implemented software artifacts [12]. This narrow emphasis on usability as a universal metric in HCI research has been criticized for not being adequate during early stages of innovations [42].

Models

Besides evaluation, HCI research has been concerned with formally describing human computer usage. While early computer systems did specific, well-defined tasks, models have been developed for specific interaction aspects such as *Hick's law* for making interaction decisions [49] and *Fitts' law* for pointing interactions [32]. As computer systems are increasingly employed to support people in higher-level activities such as learning, communicating, and decision making, a better understanding of people's tasks and their mental models in relation to computer systems becomes necessary. *User modeling* research attempts to study people using software in a more holistic way. The goal of user modeling therefore is in particular to bridge the gap between the different conceptual models of designers, programmers, and the people using the software to make computer systems more usable as well as useful [31].

Interaction Techniques

Another focus of HCI research is the design and development of interactive interfaces. Most of today's software builds on top of basic research in the realm of human-computer interaction [73]. Key techniques that have substantially influenced how we interact with information through computers are, for example, hyperlinks [74] and the direct manipulation of graphical objects as shown in early systems such as the *Sketchpad* [96] and the *Dynabook* [55].

2.1.4 Information Visualization

Information seeking can be facilitated by visualization, which typically attempts to improve the comprehension of large amounts of data or information through graphical representations. The underlying assumption of visualization is that human visual perception can be particularly helpful in discovering, reasoning, and decision making. The process of visualization consists of mapping digital representations of real world data, information, or concepts to visual representations and thereby presenting it for the perception of the human viewer (see Figure 2.4). Besides being able to influence what data is retrieved and how it is visualized, the viewer should be able to come to conclusions that are facilitated by visualizations.

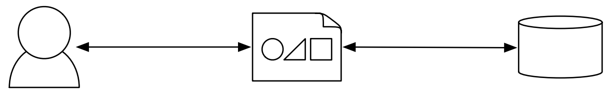


Figure 2.4: Visualization can be regarded as the facilitating intermediary between viewer and data.

Criteria for Visualizations

Whether a particular visualization is good depends on how well it represents the visualized data, how effectively it employs human perceptual capabilities, and whether it makes the most crucial aspects of the data the most accessible. Several parts of the visualization process have influence on the quality of a visualization, which can be described by expressiveness, effectiveness [63], and appropriateness [89]:

- *Expressiveness* denotes whether a visualization depicts the data in a complete and factual manner. A visualization can only be called expressive if it neither leaves out data entries nor adds elements that are not derived from the data source or a particular user selection.
- *Effectiveness* describes how well a visualization utilizes the capabilities of the output medium and the human viewer. An effective visualization would utilize the general strengths of human vision such as noticing movement and, at the same time, acknowledge perceptual weaknesses of potential viewers such as color-blindness.
- *Appropriateness* refers to the efficient use of technological resources such as computation time and memory in relation to a given goal. To design an appropriate visualization the costs and benefits of certain features need to be balanced.

Human Vision

To develop expressive, effective, and appropriate visualizations, it is crucial to consider the visual system of the human information seeker. For example, *preattentive processing* describes the ability of the human visual system to rapidly detect basic features across the field of vision. The human predisposition for preattentive processing can be utilized to enable the viewer to quickly grasp an overview of a visualization before attention has been focused on specific visual elements. Studies have shown that detecting and estimating the amount of visual elements based on their hue or orientation can be very rapid and accurate [45]. Especially for exploratory tasks, preattentive visualization was deemed very helpful to make rapid estimations and tentatively pose hypotheses. Utilizing preattentive vision requires careful considerations. For example, to support rapid detection the target visual elements have to stand out. Combining multiple possibly differentiating visual characteristics such as hue and shape might impede the rapid detection (see Figure 2.5).

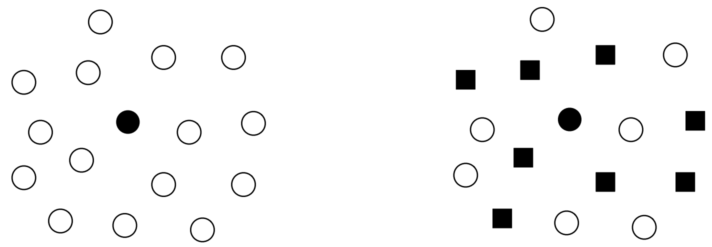


Figure 2.5: Item detection through preattentive processing: while the circle in the arrangement on the left sticks out due to a different intensity, the same circle on the right can only by detected by paying focused attention. This is due to the combination of multiple visual parameters, shape and intensity, that impede the rapid detection [45].

Visualization Pipeline

The computer-based visualization process can be described in greater detail by using the *visualization pipeline* [21], which depicts the different necessary steps for turning raw data into interactive visualizations (see Figure 2.6):

1. *Input* data is retrieved by system.
2. *Analysis* turns raw data into data structures appropriate for visualization.
3. *Filtering* reduces the number of items displayed based on criteria, possibly set through user interaction. Filtering is an optional step.
4. *Mapping* turns structured data into geometric primitives with attributes such as color, size, and position.
5. *Rendering* draws the mapped data to the output medium.
6. *Interaction* allows the viewer to participate in the visualization process by influencing how and what data is visualized.

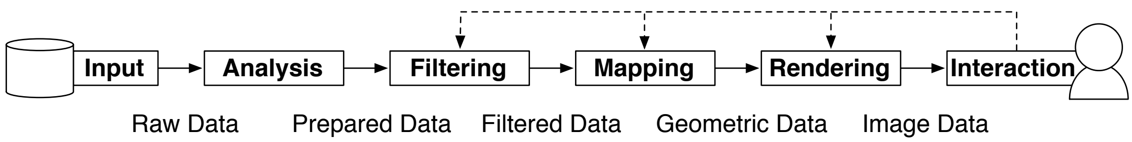


Figure 2.6: Visualization pipeline (adapted from [21]).

For static displays, visualization is a linear process, starting from data retrieval to the display and perception of the resulting image. However, interactive visualizations allow the viewer to modify certain parameters, especially concerning the selection of information items and the way these are drawn to the screen. Often it is due to the iterative, interactive visualization process that the information seeker gains new insights and makes discoveries.

Visual Information-Seeking Mantra

To design an interactive visualization system it is important to carefully conceive the tasks that should be supported. The most important visualization tasks have been summarized in the *Visual Information-Seeking Mantra*: "Overview first, zoom and filter, then details-on-demand" [91]. These tasks are part of the following seven basic tasks for information visualization:

1. *Overview.* Provide a visual overview of the data set, possibly in conjunction with a detail view.
2. *Zoom.* Zoom into a region of interest, possibly by clicking a mouse button.
3. *Filter.* Apply criteria to the visual elements to reduce amount of displayed items.
4. *Details-on-demand*: Get detailed information about individual or multiple elements, for example, in a pop-up window.
5. *Relate*. Explore the interrelations between different data elements.
6. *History*. Review and possibly undo previous interactions.
7. *Extract*. Save query parameters or selected items for later reference.

Behind each of these high-level tasks lie many techniques that allow for different ways of interacting with visualization systems. Specific visualization techniques and systems that may support information seeking will be discussed later in this chapter.

Scientific and Information Visualization

Visualization research is commonly divided into two subfields: scientific visualization and information visualization. Even though the boundaries are fuzzy, both subfields are differentiated depending on the data to be visualized. *Scientific visualization* (SciVis) is mainly concerned with the visual representation of data that has some kind of inherent spatial structure such as the human anatomy in medical visualization or the vector field in flow visualization. *Information visualization* (InfoVis) in contrast focuses on the visual representation of abstract data that usually lacks any inherent spatial structure. For example, in linguistic visualization the position and arrangement of visual elements is derived from language processing instead of a direct spatial nature of languages. Because information is abstract and typically has no explicit spatial structure, InfoVis techniques are more likely to be helpful in supporting information seeking and are thus more relevant for the work presented in this thesis.

2.2 Information Seeking Theories

Early information retrieval was mostly system-focused and assumed static information needs. This approach to information seeking has been criticized for being too constrained.

Instead, it has been argued that the information seeker should be regarded as the central component of an information system and a wide range of information seeking behaviors should be supported [14]. In this section, more elaborate descriptions and models of information seeking activity are discussed, in particular considering the cognitive, affective, and perceptual aspects.

Information seeking behavior can be described at different levels. For context and clarity the following definitions of information behaviors are adopted [108]:

- *Information behavior* describes human activity in relation to information including active and passive information seeking. Information behavior comprises any kind of human interaction with the environment, including the social context that implies communication or reception of information.
- *Information seeking* is the intended, purposeful use of a possibly computer-based information system to fulfill an information need. Examples of information systems are libraries, book shelves, and the World Wide Web.
- *Information searching* describes lower-level interactions with information systems. These information interactions are physical or mental. A physical information search task is, for example, flipping through catalogue cards, while a mental task would be the choice of a search strategy.
- *Information use* describes the physical and mental interaction with the actual information sources. Examples of information use are the annotation of a document and the incorporation of new information into existing knowledge.

This thesis focuses on computer-supported information seeking activity and the interactions that are part of it. Information seeking has been analyzed from multiple perspectives, for example, considering the technical requirements, cognitive processes, and human perceptual capabilities. The following sections discuss four approaches to computer-supported and Web-based information seeking behavior.

2.2.1 Stages of Information Seeking

In database research and early information retrieval, search has been described mostly in a system-centered way, without acknowledging many of the aspirations and problems of the information seeker. A user-centered *information search process* should instead focus on the person seeking information and his/her feelings, thoughts, and actions [59]. Studies on the process of information seeking indicated a conceptual gap between the design of information systems and the information seeker's search process. While a conventional search system assumes precision and clear organization, the information seeker's situation is typically characterized by doubts and uncertainty. Confusion about the information interest and the particular domain or topic are natural at the beginning of a search process, however, throughout the search process the information seeker's affective, cognitive, and physical activity gradually changes indicating different stages of the information search process (see Table 2.1).

Table 2.1: Information search process: the information seeker's changing feelings, thoughts, and actions indicate different stages (from [59]).

Stage	Feelings	Thoughts	Actions	Task
1. Initiation	Uncertainty	General/ Vague	Seeking Background Information	Recognize
2. Selection	Optimism			Identify
3. Exploration	Confusion/ Frustration/ Doubt		Seeking Relevant Information	Investigate
4. Formulation	Clarity	Narrowed/ Clearer		Formulate
5. Collection	Sense of Direction/ Confidence	Increased Interest	Seeking Relevant or Focused Information	Gather
6. Presentation	Relief/ Satisfaction or Disappointment	Clearer or Focused		Complete

Throughout this process, the information seeker encounters multiple information sources that need to be integrated with previous knowledge in a consistent form. The feelings of the information seeker may shift from initial uncertainty and optimism to confusion or frustration to clarity, confidence, and eventually satisfaction or disappointment. The 'formulation' stage can be considered a turning point, where the information seeker starts to clarify a previously vague information need into a well-defined search interest. From then, certainty and clarity increase from one stage to another and the information seeker's thoughts and actions become more focused. During those latter stages, most information systems are found to be more successful when the information need can be more easily expressed in explicit search terms. However, many systems do not support the tasks of the early stages, where the information seeker may feel uncertain with only a vague information need, while the information retrieval system expects precise search terms.

Information seeking on the Web does not always occur along such a linear process. Searching for information online and navigating in hypertext systems, such as the Web, have been described and modeled using food searching metaphors from anthropology and biology. Like animals forage for food, humans are described as 'infovores' relying on increasing amounts of information to make everyday life decisions. The two most important concepts for navigation in hypertext environments like the Web are berrypicking and information foraging. These two navigation concepts are described in the following.

2.2.2 Berrypicking

With *berrypicking* [13], the static concept of search separate from browsing is replaced with a more dynamic understanding of information seeking, where search and browsing are complementary and blended activities throughout the information seeking process. The searcher is described as roaming somewhat randomly between sets of documents and having always-changing information needs (see Figure 2.7). By gathering multiple information bits from search queries, the information seeker gains knowledge and thus partially satisfies his/her initial information need, which in turn blends into a new information need resulting in changing, possibly more refined queries.

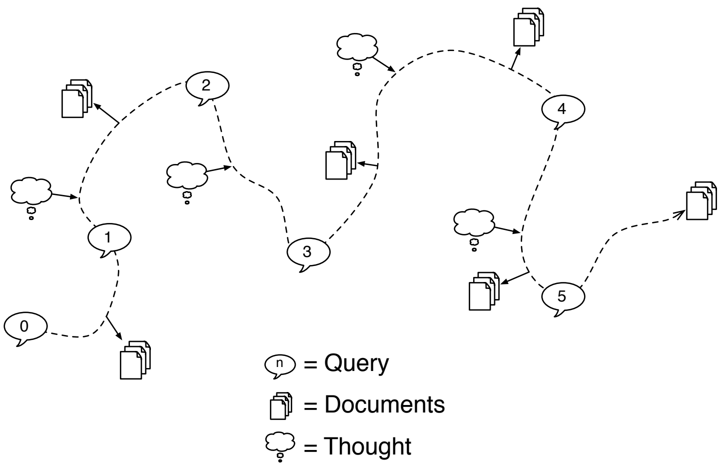


Figure 2.7: Berrypicking: roaming between document sets, evolving information needs, changing queries (adapted from [13])

An extension of berrypicking is the notion of *exploratory search* [103], which describes information seeking on the Web as often starting with a vague information need expressed with a tentative search query and followed by the exploration of search results and further queries. Exploratory search frames information seeking as a more complex activity, where a search engine lookup is merely a rudimentary subtask in the more elaborate processes of learning and investigating [67].

2.2.3 Information Foraging

Like berrypicking, *information foraging* theory [79] describes how the information seeker navigates between information resources. While focusing less on the changing nature of information needs, information foraging seeks to explain how searchers decide which information source to select and process next. An information seeker makes these decisions by rapidly estimating expected processing costs and information value through the subjective perception of visual cues. On the Web such 'proximal cues' direct attention to 'distal information' through text or graphics used as hyperlinks (see Figure 2.8).

The perception of proximal cues with regard to possibly relevant distal information is called *information scent*. Searchers rely on information scent, since search engines provide titles and snippets of results (proximal cues) helping to quickly decide which linked resources (distal information) to access. More relevant Web resources should therefore be linked with more attracting proximal cues helping the information seeker in finding

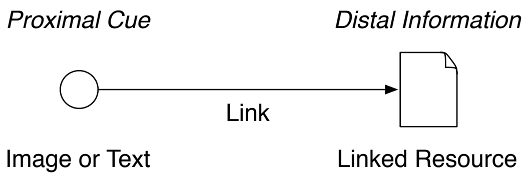


Figure 2.8: Hyperlinks as proximal cues for information scent (adapted from [24]).

the desired information. This requires a prediction algorithm that estimates the match between an information seeker's current search interest and the linked information source [24].

Both berrypicking and information foraging describe a rather low-level type of information seeking, in the sense that an overview is gained through a bottom-up process; individual resources or navigational elements guide the information seeker, but they don't provide general overviews. Only through the processing of enough information sources or cues does the information seeker gain orientation within an information space.

2.2.4 Faceted Navigation

Faceted navigation applies some of the findings of the three previously mentioned theories to the navigation of large information spaces. In contrast to conventional search interfaces that require the information seeker to formulate a query first, faceted navigation provides the searcher with textual overviews of multiple content-oriented facets that can be explored without entering explicit search queries [28]. Facets can be understood as orthogonal, non-exclusive categories that describe multiple aspects of information sources. While conventional classifications have one hierarchy, faceted classification provides multiple ways of organizing and therefore exploring information sources. A photo, for example, may be described by the place it was taken, the themes it may represent, and the people it depicts. Therefore, photo collections could be explored along these facets (see Figure 2.9). Studies have shown that information seekers would prefer faceted navigation over conventional search for exploring photo collections [112] and navigating large Web sites [29]. However, the manual creation and actualization of faceted classifications are work-intensive and require domain knowledge. By means of text-based content analysis hierarchical faceted metadata can also be generated mostly automatically [95].

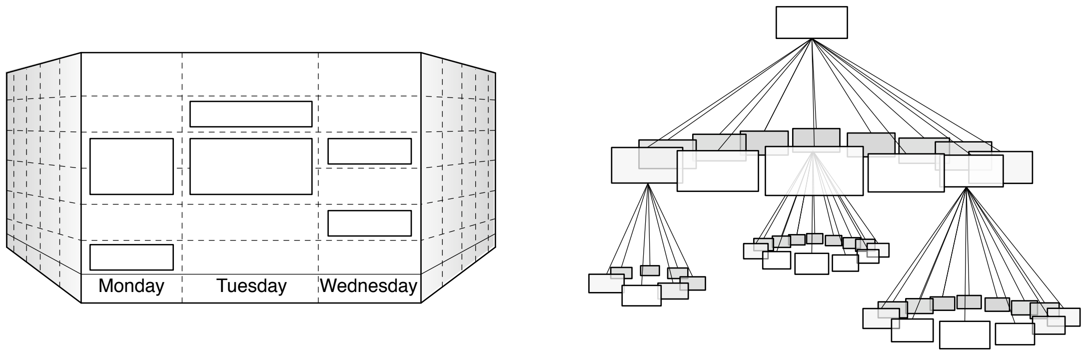
Figure 2.9: Faceted navigation allows the navigation along multiple aspects of information resources. Personal photos may be explored along place, theme, and people.

2.3 Visualization for Information Seeking

Motivated by ever enlarging information spaces, visualization research has been concerned with improving information seeking. In particular, visualization techniques have been conceived to support the information seeker in gaining an overview over a collection and navigating through it, formulating queries and assessing retrieved search results.

2.3.1 Information Workspaces

Information visualization has been seen as a method to provide a more user-centered approach towards information access. As the amount of potentially relevant information sources increases, it has been argued that lower-level information processing could provide abstraction and aggregation, which would take the cost of computer-based information seeking into account. In this sense, information workspaces are information retrieval systems which are designed to optimize the cost structure of information seeking [22].



(a) Schematic overview of Perspective Wall [64]. (b) Schematic overview of Cone Trees [86].

Figure 2.10: Visualizations for information workspaces.

Early systems that follow this approach include rather intricate mechanisms for rendering either hierarchical or linear structures into three-dimensional (3D) spaces. For example, the Perspective Wall [64] provides a focus+context environment where linear information is depicted on a plane that has a focused foreground section and two contextual side panes (see Figure 2.10(a)). Cone Trees [86] depict hierarchical structures of how information is organized in 3D, but the actual content of information sources beyond their hierarchical ordering is neglected (see Figure 2.10(b)). However, Tree-Maps [51] have shown that hierarchical data can be visualized space-efficiently in 2D, which improves labeling and thus provides more insight into the actual underlying content.

Other systems aim at improving navigation in large amounts of Web pages by providing high-level abstractions. For example, the Narcissus system exposes the hyperlink structure of visited Web pages through 3D graph visualizations [48]. The WebBook system provides different ways of navigating between variably-sized Web pages in a 3D environment [23]. Both systems feature sophisticated visualizations, however these visual representations are not based on the content of the Web pages, but their hyperlink structure or appearance. Most of these systems contribute novel ways of representing information infrastructures such as filesystems and hypertext links through a 3D visualization. The information seeker may use these visualizations to examine the structure of information spaces, while the actual meaning of individual information resources remains hidden.

2.3.2 Multiple Coordinated Views

Instead of integrating many visual variables into one complex visualization, which could lead to visual overload, *multiple coordinated views* display either different aspects of multidimensional data or use different types of visualizations. Either way, the information seeker may gain deeper insights into different aspects and interdependencies present in the underlying data. The person examining a data set through linked views can compare multiple perspectives on a given data set and may thus uncover interrelations among data entries and dimensions. Coordinating several visualizations synchronizes interactions among all views, which allows the information seeker to interact with one view while seeing effects in all linked views. For example, it is possible to zoom into a specific region of one view and the linked views would update accordingly [18]. Furthermore, temporarily highlighting items (brushing) in one view will highlight all corresponding items in the linked views (see Figure 2.11). Providing multiple views for different dimensions or zoom-levels decreases human effort of switching contexts, but it also creates computational and display-space overhead as more visualizations are generated and displayed [11]. While such coordinated visualizations can be very useful for the person exploring data, they are quite challenging to design, realize, and setup. Efforts have been undertaken to simplify 'snapping-together' multiple views [76], to formalize linking of multiple visualizations through coordination objects [17], and to visualize relations between linked visualizations [25].

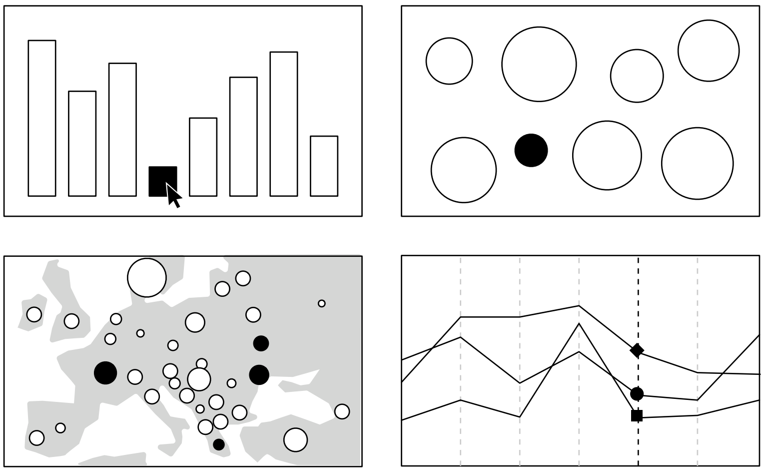


Figure 2.11: Brushing with four linked views.

2.3.3 Visual Queries

The design of early information systems assumed that the person accessing a database is familiar with the available contents, has an explicit query in mind, is well trained to use a specific query language, and sees no need to reformulate a query. However, it has been argued that most real-world information seeking tasks do not fit these assumptions [93]. Instead it is believed that semi-directed browsing, i. e., exploration, of a database or information space will help the information seeker to formulate a more adequate query and find the relevant data entries or information sources. In the following, previous research that has demonstrated how visualization techniques can improve query formulation is summarized.

Visual Data Exploration

Visual information seeking tools allow the exploration of data by the means of sliders and linked visualizations (see Figure 2.12). Instead of formulating tedious and error-prone queries in special purpose query languages, the information seeker should be able to "rapidly, safely, and even playfully explore a database" [90]. The HomeFinder system provides graphical user interface elements that enable the searcher to modify the range of several variables and thus the filtering of data entries by dragging a slider with a mouse pointer. Inspired by direct manipulation, such *dynamic queries* [4] provide a tight coupling between visual interface elements and the underlying database. As soon as the person accessing the database moves a slider, the interface is updating the corresponding amount of data entries. As an extension to the idea of dynamic queries, the Attribute Explorer provides visual overviews of variables closely aligned with interaction elements. The distribution of data entries along a dimension is displayed beyond the currently selected range. This allows the information seeker to see how changing a query may yield a satisfying set of results [93].

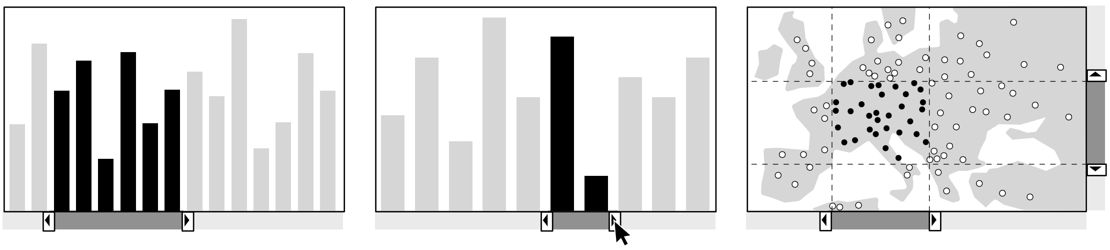
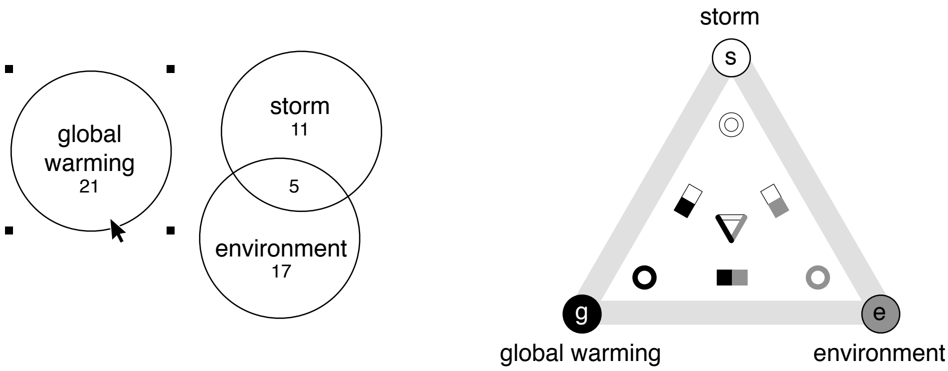


Figure 2.12: Visual data exploration: parameters can be changed using sliders.

Visual Queries for Abstract Data

The aforementioned visual query techniques allow for exploration along predefined dimensions, such as price or location, derived from a particularly structured data set. For accessing document-based information spaces, visual querying has to support abstract



(a) Query formulation using Venn diagrams. (b) Schematic overview of InfoCrystal [94].

Figure 2.13: Visual formulation of Boolean queries.

concepts. For example, the VQuery [52] system allows visual construction of Boolean multi-term queries through Venn diagrams (see Figure 2.13(a)). The authors argue that it is easier to move and overlay ellipses representing specific query terms than entering Boolean logic into form-based interfaces. However, the VQuery system does not visualize the information space or the distribution of search terms within it. InfoCrystal [94] combines visual query formulation with information visualization. The information seeker specifies a number of query terms that are put into a graphical structure, in the case of three search terms in a triangle shape (see Figure 2.13(b)). Different Boolean query term combinations are available within the triangle to allow for simple query formulation. Furthermore it is possible to combine multiple graphical InfoCrystals into hierarchical structures of filters. The information seeker explores how different kinds of filter combinations affect the results, but the visualizations within the InfoCrystals depict only how the retrieved documents relate to the entered search terms. This assumes that the information interest is made explicit and can be put into text form. Building on top of the idea of Cone Trees, Cat-a-Cone [47] provides a 3D interface for searching and browsing hierarchical information spaces. The Cat-a-Cone system is more exploratory than InfoCrystal as it enables seeing interrelations in the information space beyond currently selected categories. However, it assumes hierarchical categories for the information space, which only accommodates for specific information collections.

Visual Queries on the Web

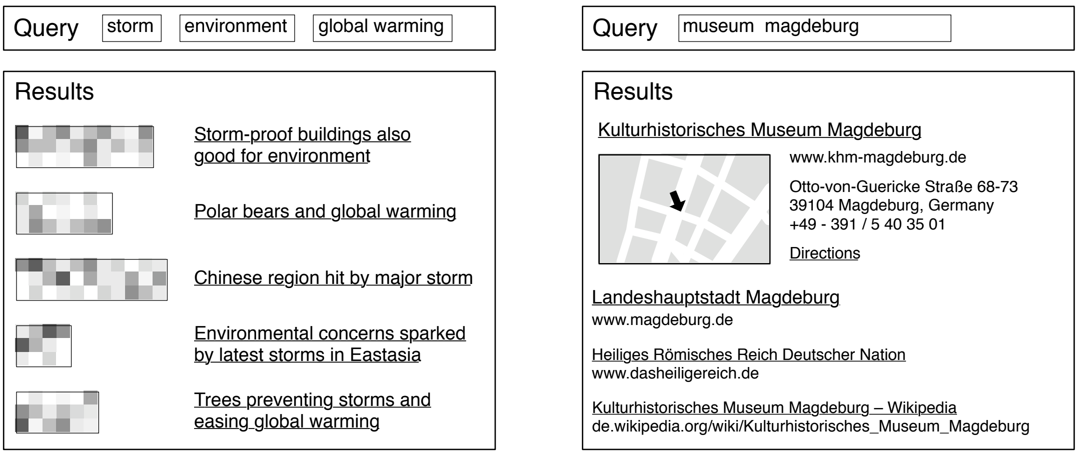
Up to now, hardly any Web-based information system has used visualizations to provide both overview of the information space and facilitate the construction of search queries. One early system features the visualization of spatial and temporal query bounds for accessing distributed geographic data from the Master Environmental Library (MEL) within the Web browser [6]. Queries are specified using conventional interface widgets such as sliders, buttons, and text fields and the resulting query bounds are displayed within the time view or geographic map by means of Java applets. However, the system does not visualize the temporal distribution of data within the time view or the spatial distribution of the data within the map. The mailing list site MarkMail [68] allows for direct temporal queries with an interactive bar chart within the Web browser. Because the interactive time visualization relies on JavaScript, third-party browser plugins are not required. Aside from MarkMail, I am not aware of any Web-based systems, where interactive visualizations have been employed to provide query formulation as well as collection overviews.

In contrast to text-based interfaces, visual queries provide a more engaging and interactive approach to accessing abstract data. However, most of these techniques were developed around the traditional desktop computer and a well-structured database or document collection assuming centralized, homogeneous, and static data. Today, the distributed nature of the Internet and the diversity of the Web require novel approaches to information exploration of networked and heterogeneous information spaces.

2.3.4 Visualizing Search Results

In addition to supporting query formulation, previous research has shown that visualization can help the information seeker in the assessment of retrieved documents. This can be done by visualizing the search results or enhancing the result list with visual cues.

The similarity between the query and the retrieved documents can be depicted as TileBars [46] showing where and how often the query terms are used in each document of the result list (see Figure 2.14(a)). This idea has been extended to scatterplots of document relevance, relevance curves, and thumbnail views [66]. Multiple techniques and representations can be combined into tables visualizing document-query similarity and document characteristics [56, 77, 100].



(a) TileBars [46] visualizing document-query similarity.

(b) Visual enrichment of search results.

Figure 2.14: Visualization for information retrieval.

The concept of information scent (see Section 2.2.3) has been applied to navigational elements to convey social relevance and support collaborative navigation [107]. Another example of information scent for navigation is the enrichment of Google's [40] search result listing with small image thumbnails, neighborhood maps, and video stills referring to possibly relevant results from Google's other services. The example given in Figure 2.14(b) shows a conventional search, with the difference that the first result item is accompanied by a map, contact information, and a link to directions. Such enhancements suggest subsequent navigation steps by embedding different types of information relevant for this particular resource.

Visual enhancements of the search results through TileBars or visual scents may ease and guide the scanning of the list. However, the information seeker still has to assess individual items in order to gain overview and orientation in relation to his/her information need. Furthermore, systems providing visualization for search results require an explicit search query at the beginning of the information seeking process.

2.4 Web-based Visualization

Early research on visualization supporting information seeking focused mostly on static data sources and local visualization logic. However, the success of the Internet and the World Wide Web demand visualization systems that work for distributed information in networked environments. The Web can therefore be seen as a medium for information visualization that can be utilized as both a data source and a delivery mechanism [87].

2.4.1 The Web as a Software Platform for Visualization

The Web browser itself may not be the most powerful platform for interactive visualizations, as the Web was originally conceived as a document-based information system. However, two important standards, JavaScript and the Document Object Model (DOM), are increasingly well supported by most Web browsers. Running in the Web browser, JavaScript provides an object-oriented scripting language that allows the interactive modification of the currently loaded Web page utilizing the DOM that provides a standardized representation of all the Web page elements. Increased support for these and other Web standards increasingly allows simple interactive visualizations [71]. For example, dynamic queries [39] can be embedded into Web pages without requiring browser extensions. The combined use of JavaScript, the DOM, and asynchronous communication with the Web server has been referred to as AJAX [35]. A particular challenge is to provide a responsive and interactive interface even though the data to be explored is distributed over the network. One approach separates the query process into two phases: query preview and query refinement [81]. Instead of retrieving all results, only statistics of the parameters are displayed during query preview, reducing network traffic and still providing collection overview. Once the result list is narrowed down to a small-enough size it can be retrieved, displayed, and further refined. Depending on the complexity of the visualization and client-side capabilities, the different processes of the visualization pipeline may run on the server or within the browser [109].

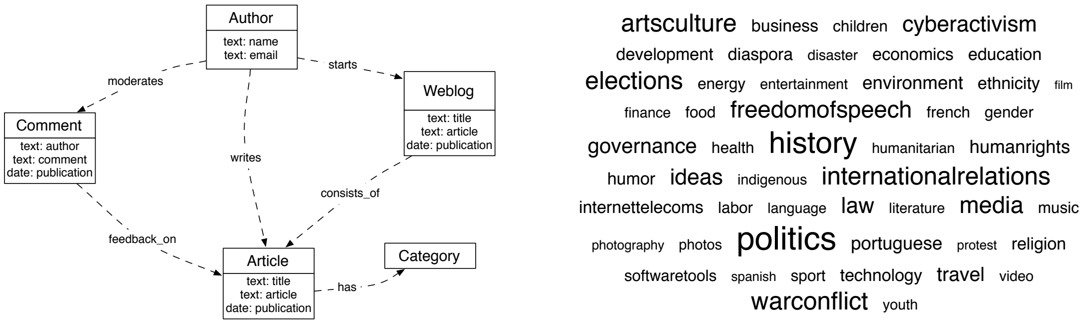
2.4.2 Social Visualization on the Web

There has been an increased interest for visualizations among ordinary people manifested by thriving discussions around Web-based visualization systems. Social data visualization describes such data analysis facilitated through visualization in social contexts on the Web. Systems like the NameVoyager [102] and We Feel Fine [43] generated considerable attention, partly due to the chosen data sets and partly because of the inviting and interactive visualizations. The NameVoyager is a Web-based visualization representing name usage over the last century, that can be filtered by entering the first letters of a name or by selecting names through the mouse pointer. This visualization was originally developed to support a book on baby name styles and choices over time. It turned out to become a great success as a facilitator for social interaction as thousands of Web users related to and socialized around the visualization even though most of them did not initially have a particular interest in baby names. While these sites are accessible on the Web, these individual visualizations are usually self-contained.

Furthermore, there are multiple Web-based applications and services that allow the non-professional visualization enthusiast to take custom data and visualize it without having to implement visualization software. Exhibit is a lightweight software for visualization of structured data within the Web browser that allows filtering and exploration of a given data set through coordinated multiple views [50]. Many Eyes is a Web community that provides a library of visualization techniques that can be used to visually represent uploaded data. Resulting information visualizations are shared and discussed among community members [101]. This way it becomes possible for non-experts to create interactive, visual representations from custom data sources and to engage in social interaction around these visualizations. However, a resulting visualization is still based on a static file. These systems don't allow for exploration or querying of a distributed information space.

2.4.3 Visualizing a More Semantic Web

As Web-based content becomes increasingly semantic and structured, visualization may constitute a more intuitive way to access growing quantities of Web-based information. While both approaches, InfoVis and Semantic Web, could considerably benefit from each other, they rely on two fundamentally different assumptions: the Semantic Web assumes data to be machine-understandable, while information visualization strives for human perception and understanding [83]. Just as some of the first visualizations for information seeking focused more on hierarchies than on the actual content, early visualizations of the Web [72] and of Semantic Web data focus on graph structures as depicted in Figure 2.15(a). Resulting visualizations may represent abstract link structures or syntactical graphs, but do not sufficiently convey overviews and patterns.



(a) Graph visualization. (b) Tag cloud.

Figure 2.15: Visualizing Semantic Web data and categories.

Tag Clouds

A simple and effective technique for visualizing free-form keywords that achieved wide usage recently is the tag cloud. Comprising a list of words in which differing font sizes represent differing frequency in use, the tag cloud indicates how often those tags were associated with resources (see Figure 2.15(b)). The concept became known through the Web communities Flickr [33] and Del.icio.us [26], where shared photos and bookmarks can be organized by free-form tags. In such tagging-based communities with many members, stable tag patterns will eventually emerge, giving way for a shared, collaboratively created taxonomy or folksonomy [38]. The tag cloud unfolded as the predominant visualization and navigation mechanism for such folksonomies as it is relatively simple to implement and gives an inviting overview of shared resources. In addition to information collections, tag clouds can also be used to provide overviews of search results [5, 60]. Studies have shown that tag clouds support a range of information seeking tasks through different ways of formatting and positioning text [85]. Several improvements have been suggested for the selection and arrangement of tags [44] and for the layout of tag clouds [54].

Geospatial Visualization

As the Web becomes also increasingly geospatial [88], geographic visualizations are providing a different, possibly more sophisticated access to resources on the Web. The visual representation of geo-referenced information through Web-based mapping systems is highly successful. One example among many is Flickr's photo map [34], which provides the world map as an exploration interface for photographs. Beyond shared content within specific communities, RSS feeds may include spatial information, i. e., longitude and latitude, or may be enhanced with geographic coordinates using a combination of natural language processor and online gazetteer [104]. Location information considerably helps during the structuring and exploration of content such as digital photos [98]. It has been shown that a large photo collection featuring both keywords and spatial information may help to label locations on a map [3]. Furthermore, the combination of tag clouds and maps in one view has been proposed to support initial exploration of large, multi-faceted data sets [92, 110]. However, the blending of multiple dimensions into a merged view creates particularly dense representations which may be difficult to examine and explore.

Web-based mapping and tag clouds constitute effective and appropriate information visualizations for the Web, as they make good use of the Web browser and the human visual system with reasonable computational effort. However, maps and tag clouds individually are not fully expressive visualizations, because they cannot sufficiently represent the multiple aspects of Web-based information. The approach of merging multiple visualizations into one view appears to be problematic, since the combination of overlapping visual variables (e.g., size and position) representing multiple aspects (e. g., location and tags) may cause visual overload.

2.5 Summary

This chapter gave an overview of the research that forms the foundation for the work presented in this thesis. At first four research areas within computer science were presented that contribute to the study of information seeking: information retrieval, World Wide Web, human-computer interaction, and information visualization. Then an overview was given of theories describing information seeking activity especially concerning its cognitive, affective, and perceptual aspects. After that visualization research that seeks to improve information seeking was presented, with particular attention given to information navigation, query formulation, and the presentation of search results. Finally, it was discussed how the Web evolves into an InfoVis medium that is ready to be utilized as a software platform and a community space for visualization. In the following chapters, I develop the idea of visual information exploration on the Web.

3 VIEW: Visual Information Exploration on the Web

In this chapter, I develop the idea of visual information exploration on the Web (VIEW). First the limitations of current information seeking theories and visualization techniques are briefly summarized (Section 3.1), then, on the basis of previous work, I introduce the VIEW concept (Section 3.2), and finally I compile specific design goals for visualization widgets that follow the conceptual requirements of the VIEW approach (Section 3.3).

3.1 Problem Analysis

The Web is evolving into a universal source of information, with enormous dimensions. This context requires better exploration mechanisms for information seekers. Two current trends suggest that it may now be more possible to apply visualization to information seeking on the Web: 1) Web-based information is becoming increasingly structured, or can be turned into a more structured form, and 2) today's Web browsers allow for richer interaction without requiring specific browser plugins to be installed. Both more structured Web resources and better interactivity within the Web browser make visualization for information seeking on the Web more feasible.

Information visualization research has been predominantly concerned with making large quantities of abstract data more accessible through interactive, graphical representations that provide overview, filtering, and exploration functionality [91]. Since its early days of existence the World Wide Web constitutes a challenging 'data set' within InfoVis research. Many systems and techniques relate to the Web with the goal to provide visual access to such an extraordinarily large and increasingly important information space. However, up to now visualization has been rarely utilized to support information seeking on the Web. Searching and browsing on the Web remain rather low-level processes, visualization systems mostly concentrate on constrained data sources, and the Web's potential as both visualization platform and information space is seldom acknowledged.

3.1.1 Information Seeker and System

Information-seeking theories describe the use of technology to access information sources and to eventually fulfill information needs (see Section 2.2). Previous research on information seeking has put differing degrees of emphasis on the person seeking information and the software supporting the information seeker. On the one hand, information retrieval is a more technology-focused approach to information seeking, where algorithms and data structures are conceived to support the retrieval of information sources. The needs of the information seeker are merely represented by search queries and possible relevance feedback. On the other hand, theories such as the information search process [59] describe the changing feelings, thoughts, and actions of the information seeker, while the role of the system throughout the different stages is not addressed. However, neglecting the strengths of either the information seeker or the information system may impede the overall information seeking process, since it depends on both.

Approaches such as berrypicking and information foraging describe the relation between information seeker and system as being more interdependent. Berrypicking [13] models an ongoing interaction between system and user, in which the user's cognitive processes and the documents provided by the system both influence the progress of the information-seeking process. However, the interaction between the person searching for information and the system providing information is limited to exchanging queries and results. Information foraging [24, 79] examines how the information seeker selects or discards information sources and applies this insight to the design of information systems. Through a prediction algorithm the system may provide visual cues for more relevant information sources, which would utilize the perception system of the information seeker. The resulting information scents guide the information seeker between information sources, yet visual overviews of the whole information space are not provided.

While the relation between searcher and system is increasingly framed as being interdependent, the strengths of both may not be sufficiently addressed in current information seeking systems. To improve information seeking, the searcher should be better supported in performing high-level tasks such as learning and reasoning, while the information system should implement the necessary mechanisms. In particular, the interactions between system and searcher need to be improved and the information seeker should be provided with overviews of the information space.

3.1.2 Low-level Information Seeking

An important part of information seeking is developing an understanding of available content in general as well as in relation to an information need. Gaining overview and orientation on the Web requires the information seeker to formulate multiple queries, evaluate many retrieved documents, and navigate along hyperlinks. While this process is highly interactive, it can be described as low-level and bottom-up information seeking, since many individual search results and Web pages have to be accessed and evaluated to develop an approximate orientation within the information space. First, the information seeker has to turn an initially vague information need into an explicit set of search terms; then the searcher has to scan individual search results to decide on the next steps, either looking at some of the returned items in detail or changing the search query.

I criticize this low-level information seeking process as being often laborious, since the information seeker has to formulate multiple queries and consider many individual information items to gradually gain overview and orientation in the information space. This, however, may turn out to be problematic as the number of information sources steadily increases and thus also the amount of possibly relevant information. Furthermore, many information systems constrain the information seeking process by requiring explicit search query terms. However, studies indicate that information needs early in the information seeking process are relatively vague [59].

Exceptions to this low-level type of information seeking are faceted navigation, visualizations of search results, geographic maps, and tag clouds. Faceted navigation provides textual overviews, however, it does not visualize the information space along these facets. Visualizing or visually enhancing search results improves the process of assessing the relevance of returned documents, yet this requires the information seeker to turn an implicit information interest into a formal search query, which may constrain the information seeking process. Maps and tag clouds provide opportunities for exploration and higher-level overviews of particular aspects of an information space, yet when they are used individually they are limited to a specific dimension, e.g., location or topic.

3.1.3 Data Assumptions

The Web is a particularly interesting and challenging data set for InfoVis research due to the large quantities of distributed, dynamic, and diverse information items. However, most of the systems that attempt to improve information seeking through visualization are constrained by structural dimensions or a limited scope to a particular data set.

Structure

Visualization systems that aim at providing abstraction of an information space usually rely on one of the following structural entities that are chosen as dimensions for information seeking. Even though these structural dimensions are assumed to be meaningful for the information seeker, they usually imply certain limiting assumptions:

- *Query terms or database fields.* Visual query techniques only provide graphical representations in relation to the search query terms [94] or specific database fields [4, 90]. This limits the exploration space for the information seeker.
- *Hierarchies.* Information visualization based on categorical hierarchies [47, 86] requires elaborate efforts for maintaining the categories or, when derived from storage structure, may not adequately represent the contents of individual information sources.
- *Thumbnails.* Thumbnails are icon-sized images of information sources that may help to recover previously visited information items such as Web pages [23]. While thumbnails indicate the visual appearance of a Web resource, the context and content of textual resources may not be sufficiently conveyed by the means of thumbnails.

- *Link structures.* Hyperlinks on the Web denote relations between Web resources, however, the meaning of these relations can be rather ambiguous. Direct visualizations of link structures [48] may therefore not be sufficiently expressive or effective for representing the content of information sources. Content analysis of interlinked resources may be necessary to improve link-based visualizations.

- *Single dimensions.* Web mapping and tag clouds provide visualizations for particular dimensions of information sources. Yet individually, location and topics may not be sufficient for exploring an information space. Merging tags and maps into one combined view is problematic due to probable visual overload.

These above-named structural dimensions constitute starting points for most information visualization systems. However, these dimensions are rather system-centered and confined, since they are either mostly derived from storage and data structures or limited to a particular dimension. While they can be effectively and appropriately used for visualization, employing any of these individually may not sufficiently fulfill the expressiveness criterion. Basing visualizations on multiple, more user-centered dimensions may help represent more of the multiple aspects of the Web's diverse information sources and perhaps be a step towards supporting the diversity of human information needs.

Scope

Most visualization systems have been developed with a focus on one of the following data sets that are usually local or static and thus do not account well for the distributed and dynamic nature of the Web:

- *File.* Visualization of individual files allows exploratory data analysis, but it excludes information seeking by definition.

- *Database.* Providing visual access to a database improves data exploration, which is usually constrained by the database fields.

- *Web community.* Visual exploration techniques such as maps and tag clouds have been employed for larger amounts of Web resources within online communities. These developments greatly extend the use of visualizations for information seeking, yet, typically these visualizations are limited to the information space of a specific community.

Concentrating on information coming from a well-defined context makes the data more predictable and thus allows for less complex analysis and mapping steps. Nevertheless, limiting the scope of a visualization also limits its usefulness for information seeking in larger information spaces.

3.1.4 Data Source or Software Platform

Considering visualization and information seeking, the Web can constitute two things: software platform and data source. Early visualization research has already addressed the Web as an information space that can be visualized and increasingly the Web is also used as a platform for information visualization. Improving client-side interactivity and emerging semantics suggest considering the Web as both a software platform and a data source. However, only few systems embrace those two roles of the Web, which may impede the role of visualization for information seeking.

3.2 Concept of VIEW

With *visual information exploration on the Web* (VIEW), I approach information seeking by building on top of and extending the theories and techniques discussed in the previous chapter. Facilitated by interactive visualizations, VIEW allows visual information exploration that embraces the Web as an information space and as a software platform. In the following, I briefly describe the goals and corresponding requirements of the VIEW approach.

3.2.1 Visual Information Exploration

Currently information seeking is often a low-level and constrained procedure. In contrast, VIEW is a concept that aims to embody visual information exploration as high-level, engaging, and fluid information seeking:

- **High-level views**

  In a VIEW system the information seeker is offered high-level perspectives on an information space through overview visualizations that abstract and aggregate multiple dimensions of the information sources. By utilizing preattentive cues to better enable rapid comprehension of visual overviews, the intention is to support the information seeker in surveying relevant items or possibly more easily encountering interesting topics.

- **Engaging interactions**

  In addition to looking at visual overviews, the information seeker can engage interactively in navigating through an information space. As opposed to conventional searching or browsing, VIEW provides information interaction by means of direct manipulation of graphical elements embedded into visualizations.

- **Fluid changes**

  Exploring information spaces with a VIEW system involves progressive refinement of query parameters. Any interaction undertaken with visual query tools in the VIEW system leads to comprehensible visually displayed changes of the interface.

All changes triggered by every gradual adjustment of the visual query parameters is immediately shown visually in the amount of retrieved information and other interface aspects.

Previous techniques for information seeking and visualization placed emphasis on either the retrieval or the visual representation of information sources. The intention of the VIEW concept is to combine interactive visualization and retrieval tools to better enable higher-level, more engaging, and more fluid information seeking (see Figure 3.1(a)).

With VIEW I propose a more complementary relation between the system and the person during the information seeking process (see Figure 3.1(b)). This idea is related to the notion of *man-computer symbiosis* [62] that assigns low-level routines to computers and high-level activities to humans. In the case of VIEW, low-level tasks such as data aggregation, abstraction, and mapping are carried out by the system to support the information seeker in higher-level tasks such as learning, exploring, and reasoning.

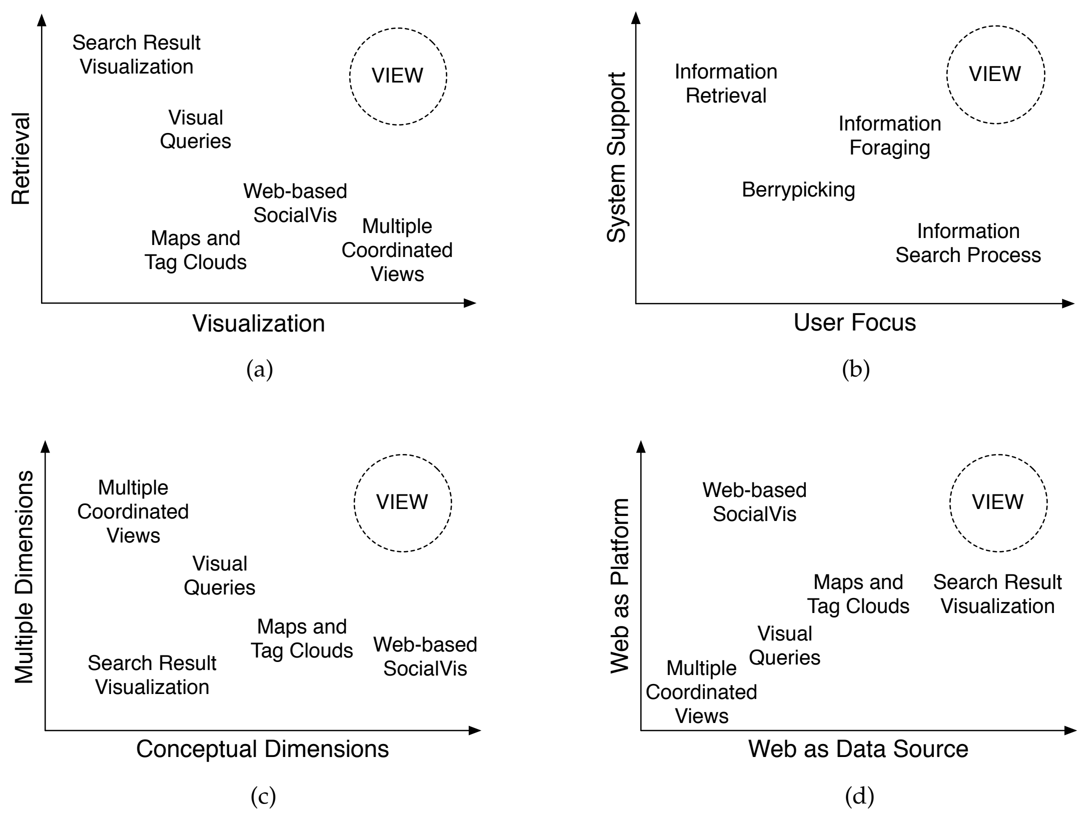


Figure 3.1: Simplified graphical comparisons between the VIEW concept and related theories and techniques.

3.2.2 Multiple Conceptual Dimensions

Structural dimensions such as database fields or link hierarchies describe technical conditions, but they may not adequately represent the actual content of information sources. For example, Web resources are unique in their interlinked structure, but they have many more facets that cannot be sufficiently captured by link graphs alone. To improve visual information seeking, information needs should be described by multiple concepts that are based on mental models of information seekers and that are not constrained by specific data types or dimensions.

With VIEW I explore a shift away from system-centered structural dimensions towards more user-focused aspects by choosing conceptual dimensions as the basis for visualization and information seeking. A *conceptual* dimension refers to one aspect of an information space that is intended to be meaningful and significant for the information seeker. Instances of this conceptual dimension can be derived from structural entities available within the information space. Structural entities such as timestamps or pairs of latitude and longitude serve as low-level representations that can be converted into more accessible, conceptual representations, for example, the publication date and the geographic origin of an information source.

While individual conceptual dimensions may not be fully sufficient for visual information seeking, a combination of *multiple* conceptual dimensions may add more expressiveness, especially if they are relevant to the information seeker and suitable for the information space. The VIEW system is therefore carefully designed to include multiple conceptual dimensions (see Figure 3.1(c)).

3.2.3 Information Space and Visualization Platform

Most InfoVis systems either visualized the Web as a data source or utilized the Web as a platform. However, the Web's recent advances—in particular emerging semantics and improved client-side interactivity—suggest the possibility of a more holistic approach to the Web. VIEW aims to extend the role of the Web through information seeking by visualization (see Figure 3.1(d)). A VIEW system should allow both exploration *of* the Web as an information space as well as interactive visualization *on* the Web as a platform. In this section, I discuss considerations and implications for VIEW systems regarding these two roles of the Web.

The Web as a Data Source

VIEW assumes an information space to be distributed, dynamic, and diverse as the Web. In contrast to many visualization systems, a VIEW system therefore has to account for the aggregation, updating, and integration of information sources. The VIEW approach can be applied to differently sized information spaces, ranging from a personal set of bookmarks to a search engine's collection. In the following, I discuss three major types of Web resources and their potential for VIEW.

- **Hypertext**

  As the WWW's original document format, HTML is naturally the most predominant format on the Web. The syntactical structure of an HTML file corresponds to its layout and presentation rather than its semantic content. There are multiple extensions that may embed semantic information, but they are employed neither widely nor consistently. A reasonable amount of pre-processing and inference mechanisms are necessary to derive conceptual dimensions, monitor changes, and integrate diverse information sources.

- **Syndicated content**

  Syndicated data such as RSS feeds (Really Simple Syndication) constitute a widely adopted format for distributed information that is primarily used to convey updates and changes on Web sites, such as blogs, online communities, and news sites. These feeds typically do not include presentational information, but are semantically structured in a rather consistent manner allowing for easier processing. While there are slightly deviating versions of RSS feeds, all have a well-defined structure that essentially includes the same basic dimensions: title, publication date, description, a link, and categories. Multiple extensions allow more sophisticated metadata for media annotations and geographic locations. By design, syndicated content provides a better infrastructure for aggregating, integrating, and tracking information sources.

- **Semantic Web data**

  Data formats such as RDF (Resource Description Framework) allow near-universal information representation with a wide range of syntactical structures. In comparison with hypertext markup and syndicated content, Semantic Web data provides the most sophisticated data and information modeling. While the low-level visualization of semantic data is rather straightforward using network graphs, semantically appropriate visualizations may require domain-specific knowledge. The aggregation, integration, and monitoring of diverse Semantic Web resources requires elaborate mechanisms and domain knowledge.

Beyond these three example data types, the Web features numerous additional document and media types (e. g., PDF), data structures (e. g., databases), and special domain syntaxes (e. g., Wikipedia). Typically there is some kind of structured information embedded in those information sources that can be transformed into conceptual dimensions (see Section 3.2.2).

The Web as a Platform

VIEW systems rely on sophisticated interaction, visualization, and data processing and the Web has evolved into an appropriate architecture for this. Being accessible on a wide range of devices—from personal computers and laptops to PDAs and cellphones—the Web constitutes a platform with probably the widest distribution. In the following, I discuss the implications of utilizing the Web as a platform for VIEW:

- **Immediate Access**

  While previous Web visualizations have been separated from the Web-browsing experience, visual information exploration should be embedded in the actual browsing context, i. e., the Web browser. This way information seekers do not have to change contexts, since the Web browser already constitutes the major software for information search and exploration online. Because a VIEW system is Web-based software it has the advantage that it can immediately be accessed by information seekers from within their Web browsers. There is no need to install or configure any software; an URL or a keyword is sufficient to start exploring information online.

- **Context for Conversation**

  Today, the Web is the predominant platform of digitally mediated community and conversation. Apart from email and instant messaging, millions of people gather and socialize in Web-based online communities such as Facebook and MySpace. Web communities are also often based around the notion of sharing resources such as photos or bookmarks. VIEW has the potential to become part of online conversation, if it allows the information seeker to share findings via online communities, email, instant messaging, and blogs.

- **Established Architecture**

  Developing interactive visualization software for the Web requires a range of programming languages and development contexts. This is due to having both the Web browser and the Web server provide functionality. While this causes a more complex development process, it also allows a more sophisticated allocation of responsibilities along the visualization pipeline. As a powerful information repository with a database, the Web server handles data analysis and filtering before any data is sent over the network and the Web browser takes care of mapping and rendering according to the client's display capabilities. This allows for visualization of large volumes of Web resources and limits the processing time on the client.

Being immediately accessible for most computer users, allowing community and conversation, and having an established client-server architecture, the World Wide Web constitutes a compelling platform for visual information exploration.

3.3 VisGet Design Goals

The basic idea of the VIEW approach is Web-based information exploration facilitated through interactive visualizations offering both visual overviews and query tools. These interactive visualizations can also be described as *InfoVis widgets* (VisGets) combining visual representation and retrieval of Web resources. In the following, the design goals of VisGets are outlined.

Broadly speaking, the goal of VisGets is to expand the possibilities for formulating queries and accessing Web resources within the Web browser. More specifically, VisGets may particularly ease the specification of some concepts that are difficult to express in text. For example, time is difficult to specify in text without using several words: a time range such as "last summer" or "during February or March last year" takes several words, and might not be particularly effective as a query in a text-based search engine. However, time as a concept can be highly useful for humans as a query filter.

In order to be more precise about the design goals for VisGets, the terms that are used to describe them are defined. For this discussion, a *query* is a user request for information from a large information space. A query is composed of a set of *parameters*. A parameter is a piece of information within a query that represents a part of an information need or interest. This parameter can be thought of as a constraint by which the entire information space is filtered. It can also be thought of as an attractor for information items based on similarity factors. Commonly, in Web searches, these query parameters are words and complex queries are built using Boolean combinations of words.

Query parameter *dimensions* are an important feature in VisGets. A query parameter dimension is a concept that on the one hand is meaningful for the information seeker and on the other hand is prevalent enough in the information space to be effective as a filter. For instance, photos are increasingly associated with timestamps, geographic coordinates, and keywords. Particularly useful query parameter dimensions could be events, places, and themes, with which one should be able to explore photos. In making a query based on a dimension, one manipulates a *range* rather than using a word or set of words. A range defines a subset of the information space along a particular dimension.

These are the eight specific design goals for VisGets:

1. **Enable casual formulation of complex queries**

   A VisGet should support casual exploration of large information spaces using sophisticated queries. Complex queries should be constructed in conjunctive form (e. g., Boolean AND) using a combination of multiple query tools. VisGets should particularly allow the formulation of search queries based on parameters that are difficult to specify with textual queries.

2. **Summarize information collections visually**

   The display should include a visual overview of the parameter dimension in the VisGet, as well as a clear indication of the currently selected and filtered items. The interface should allow interactive exploration and review of the interrelations between multiple query dimensions.

3. **Visualize bounds of query dimensions**

   The information seeker should be able to view the currently selected range for each query dimension, as well as the full range available for each query dimension. It should be easy to adjust the selected range, and to switch between a selected range and the full view.

4. **Visualize query changes**

   As query parameters are modified in one VisGet, the effect of these changes should be displayed within this VisGet itself, and simultaneously reflected in all the other VisGets. The set of information items in the results list may be updated incrementally, using transition animations to help the information seeker understand the incremental changes to the information set.

5. **Use integrated dynamic manipulation**

   A VisGet should provide interaction methods for adjusting the parameter range, and should provide responsive updates for the results of query adjustments. The interface should support interactive exploration of the relationships between multiple query dimensions.

6. **Provide information drill-down**

   The interface should provide access to appropriate resolution in a VisGet's parameter dimension. The information seeker should be able to display detailed information for result items upon request (e. g., details-on-demand or detail-in-context).

7. **Provide interaction history**

   Interactions with VisGets should be recorded in an interaction history that provides functionality for retracting any parameter changes. The information seeker should be able to revoke the last interaction or easily go multiple steps back. Specific selections can also be saved for later reference.

8. **Enable lightweight information sharing**

   The information seeker should be able to share a query, i. e., VisGet parameters and the selected information, with other people. Such query sharing should be simple and lightweight.

These goals are embodied by the VisGets that are part of an implemented VIEW system as described in the following two chapters.

3.4 Summary

In this chapter I introduced the idea of visual information exploration on the Web (VIEW) by formulating a problem analysis of current visual information seeking, introducing the VIEW concept, and presenting design goals for VisGets. As the Web evolves into an infrastructure for information, community, and software, information seeking theories emphasize either the user or the system, which may be part of the reason why current information seeking is low-level, laborious, and constraining. Currently, most information visualizations make limiting assumptions about data sets, and fail to consider the Web as both information space and software platform. From these points of critique I developed the VIEW approach that comprises three main principles:

- Visual information exploration should be a high-level, engaging, and fluid process that builds on top of a complementary notion of an active information seeker and a sophisticated information system.
- Multiple conceptual dimensions should provide a more meaningful and expressive access to an information space than individual structural data dimensions.
- The Web should be embraced as both information space and visualization platform that allows for immediate access, community and conversation, and sophisticated software.

Derived from this conceptual discussion, eight specific design goals for VisGets were outlined, providing a blueprint for the realization of a prototype system presented in the next two chapters.

4 VisGets: Coordinated Visualizations for Information Exploration

Conceived to enable the VIEW approach, VisGets are an attempt to support visual information exploration on the Web by means of interactive visualization widgets. This chapter presents the design of VisGets that are part of an implemented VIEW system, which will be explained in the next chapter. First the choice of three conceptual dimensions for visual information exploration is discussed (Section 4.1). Then Section 4.2 describes the design of three initial VisGets with respect to their appearance and functionality. Section 4.3 briefly discusses how VisGets are embedded into a VIEW interface that combines the VisGets with search and results elements. Two types of coordinated interaction with VisGets are explained in Section 4.4. Finally, the use of VisGets within the Web browser is examined in Section 4.5.

4.1 Choosing Information Dimensions

The Web is a source of large volumes of diverse information that, as it evolves, continually requires new approaches for information exploration mechanisms for information seekers. Recent developments indicate that Web-based information is increasingly structured, which makes the Web a fertile domain for more sophisticated queries and visualizations. Increasingly, information resources published on the Web are often organized along three main dimensions: time, location, and tags. While there are many other dimensions used to categorize and structure information on the Web, these three dimensions have been selected as examples that are widely used, meaningful, and fairly easy to extract. In the following these dimensions are briefly discussed in regard to their relationship to visual information exploration.

4.1.1 Time

Time is a dimension that is universally employed for computer-based information and is also relevant for human information seekers. Practically every resource on the Web has at least some kind of temporal information associated with it, indicating its time of creation, publication, or modification. This universality of the temporal dimension makes it an effective instrument for organizing and exploring digital resources. Examples of Web-based resources that are primarily organized by time are blog entries, recent changes on Wikis, and news feeds from friends on social networking sites. These mechanisms keep the information seeker up-to-date on current developments, either globally or within a closer social context. In general, information resources can be contextualized along the time dimension in relation to other resources that have been published at the same time. Allowing the exploration of past conversations or presenting what is being discussed in a given moment makes time a particularly significant dimension for the VIEW approach.

4.1.2 Location

In the past, location information was not widely used to structure computer-based information, even though it is a significant dimension for human reasoning and discourse. However, with built-in GPS (Global Positioning System) capabilities in many of our digital devices such as cameras and cellphones, more and more information published on the Web has geospatial information attached to it. Photos, news items, and even encyclopedia entries increasingly include geographic longitude and latitude. Even information that has no explicit location information can be enhanced with pairs of longitude and latitude using natural-language processing in combination with geographic look-up services [104]. Location constitutes an especially meaningful information dimension as it provides spatial contexts for usually abstract information resources.

4.1.3 Tags

Topics and themes associated with Web-resources are often made explicit through free-form keywords or tags. Especially in communities, where resources such as photos are shared, free-form tags provide an organization scheme with little overhead for the person sharing resources. While time and location are more contextual aspects of information, tags can concisely represent the content of information resources. Apart from neutral content descriptions, tags can also convey contextual information such as opinions and feelings, since tags can typically be applied to shared resources by people other than the publisher. As tags can be categories, keywords, and subjective statements, tags provide a rich and diverse semantic dimension for information exploration.

4.2 Appearance and Functioning of VisGets

A VisGet is an information visualization widget that combines visual representation and interaction for a particular information dimension. The visualization part of a VisGet can be a simple visual overview of one aspect of an information collection. Interaction with a VisGet solely depends on direct manipulation using the mouse pointer. Depending on the visual representation, interactive elements are embedded into the widget and provided as separate controls. VisGets follow a consistent layout and color scheme (see Figure 4.1). In the upper left corner the name of the dimension is displayed. A VisGet can be reset to defaults by clicking 'x', which is situated to the immediate right of the dimension title, if the VisGet is actively filtering. Besides the reset control there can be additional controls depending on the VisGet. Interactive elements can also be embedded into the visualization area. Consistent use of colors indicates the roles of visual elements. Green controls allow modification of the presentation properties of a VisGet, while orange controls allow altering query bounds. Items in the visualization can be blue, pink, or gray: blue items are currently selected, pink items are temporarily activated, i. e., brushed, and items beyond the query bounds are displayed in gray. Visual items representing information sources also function as controls, as they can be selected to narrow the query bounds of the VisGet around the value of the particular item.

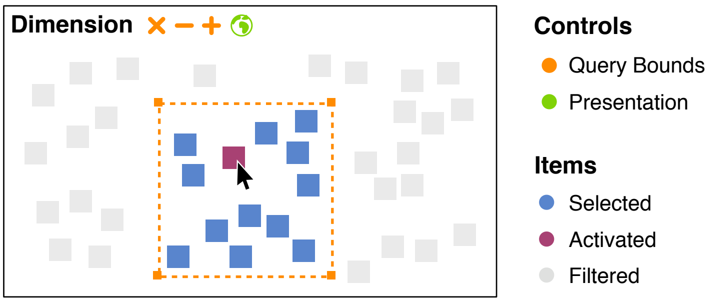


Figure 4.1: VisGet layout and color scheme.

In the following, I explain in detail how VisGets work by presenting three initial VisGets for time, location, and tags. The information resources that are used for the given examples are taken from RSS feeds of Global Voices Online [37], an editorial aggregation Web site about blogs around the world.

4.2.1 Time VisGet

For the temporal dimension, a simple bar chart indicates the range of months that can be selected and how many information items have been published on a monthly basis (see Figure 4.2). The temporal selection can be changed by dragging the sliders that are located below the bar chart along the horizontal axis. An individual month can be selected by narrowing the sliders accordingly, or by clicking on the month bar itself. An additional bar chart representing days of an individually selected month is shown, allowing filtering at finer granularity.

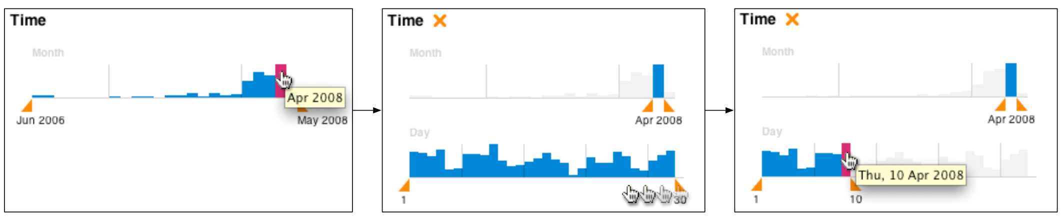
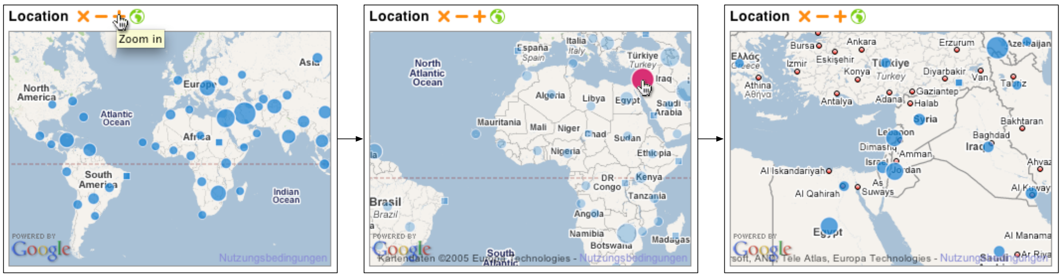


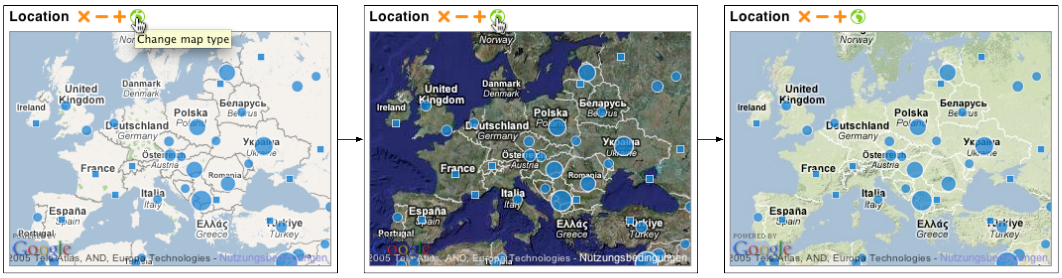
Figure 4.2: Refining along the temporal dimension using the time VisGet: temporal ranges can be changed by dragging the orange sliders or by selecting individual months or days.

4.2.2 Location VisGet

The location VisGet includes a geographic map, upon which superimposed squares and circles represent information items (see Figure 4.3(a)). Squares mark the location of individual items, while circles represent aggregated items, with the area of the circle reflecting the number of items contained. The spatial query parameters, which are embodied by the map's display boundaries, can be changed by zooming and panning the map. Zooming in and out can be performed via the scroll-wheel, double-clicking either the left or the right mouse button, or by means of the '+' and '-' buttons at the top of the VisGet. Furthermore, it is possible to select circles or squares and zoom into the map to show a more detailed spatial distribution of the region containing selected information items. As geographic maps can be visualized based on different data, for example, political boundaries, satellite imagery, and elevation, the earth control allows changing the map type used for the location VisGet (see Figure 4.3(b)).



(a) Zooming by using the zoom control and by clicking on circles.



(b) Switching between map types using the earth control.

Figure 4.3: Interactions with the location VisGet: (a) changing the query bounds and (b) changing the type of presentation.

4.2.3 Tag VisGet

The tag VisGet features an alphabetically sorted tag cloud giving a topical overview of the information collection. The font size of each tag represents how often it appears among the information items. The overall range of the topical dimension is based on the tags used within the whole information collection, while the bounds for a particular query are set through individual tags. A tag can be selected as a query parameter by clicking on it, after which the tag background turns orange (see Figure 4.4). Multiple tags can be selected as filters concurrently, which are then combined to a Boolean conjunction: the more tags are set, the more information items are filtered out. Hovering the mouse pointer over a tag will highlight related tags. This behavior will be discussed as brushing in Section 4.4.1. Visual overcrowding of the tag cloud is avoided by limiting the quantity displayed to the most frequently used tags. Seldom used tags can be shown by either selecting the '+' button on the top of the tag VisGet or by refining the tags selection or by setting query parameters in other VisGets.

Figure 4.4: Selecting the tag 'artsculture' as a query bound for the tag VisGet.

4.3 VIEW Interface

VisGets are an integral part of visual information exploration. The interface of the implemented VIEW system contains the three aforementioned VisGets and a search query box in the upper area and the results in the lower part (see Figure 4.5). Results are displayed depending on the query parameters specified through VisGets and search query.

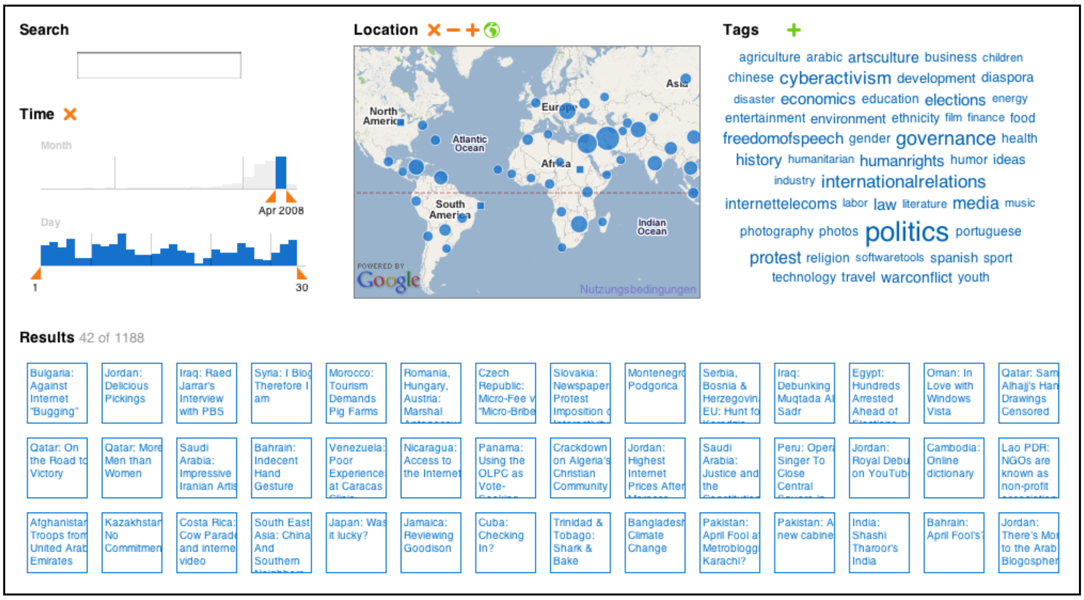


Figure 4.5: Interface of the implemented VIEW system.

4.3.1 Search Query

Besides specifying query parameters using VisGets only, it is possible to combine the query parameters of the VisGets with a full-text search using a conventional text query box. The search query of the VIEW system behaves similarly to the VisGets as it becomes orange when active and is removed from the current query formulation by selecting the reset control 'x'(see Figure 4.6).

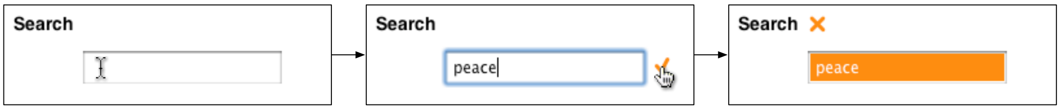


Figure 4.6: Searching for information sources containing the word 'peace'.

4.3.2 Result List

The result list depicts information items that comply to query parameters. Results are displayed as squares with the title of the information source embedded. When queries change, information items are removed and added through animated transitions. Each result item constitutes a hyperlink leading to the actual information source. Hovering over a result item shows a preview of its description, when, by whom, and where it was published (see Figure 4.7). Images potentially included in information items are displayed in the upper right corner of the detail overlay.

Figure 4.7: On demand detail information in the result list.

4.4 Coordinated Interaction with VisGets

While individual VisGets allow visualization and interaction along one dimension, several linked VisGets provide functionality for multi-dimensional query formulation and linked visual overviews. The implemented VIEW system provides two types of coordinated interaction with multiple VisGets: weighted brushing and query refinement.

4.4.1 Weighted Brushing

Hovering with the mouse-pointer over a visual element in one single VisGet temporarily highlights related visual elements in all linked VisGets. This selection is immediately dismissed as soon as the mouse-pointer moves beyond the edge of the visual element. Highlights appear in a different color than the items that are not activated. For example, hovering over a tag in the tag VisGet highlights all related elements in the temporal bar chart, geographic map, and result list (see Figure 4.8). In the implemented system highlighting is done by changing the color of visual elements from blue to pink, and dimming unrelated items.

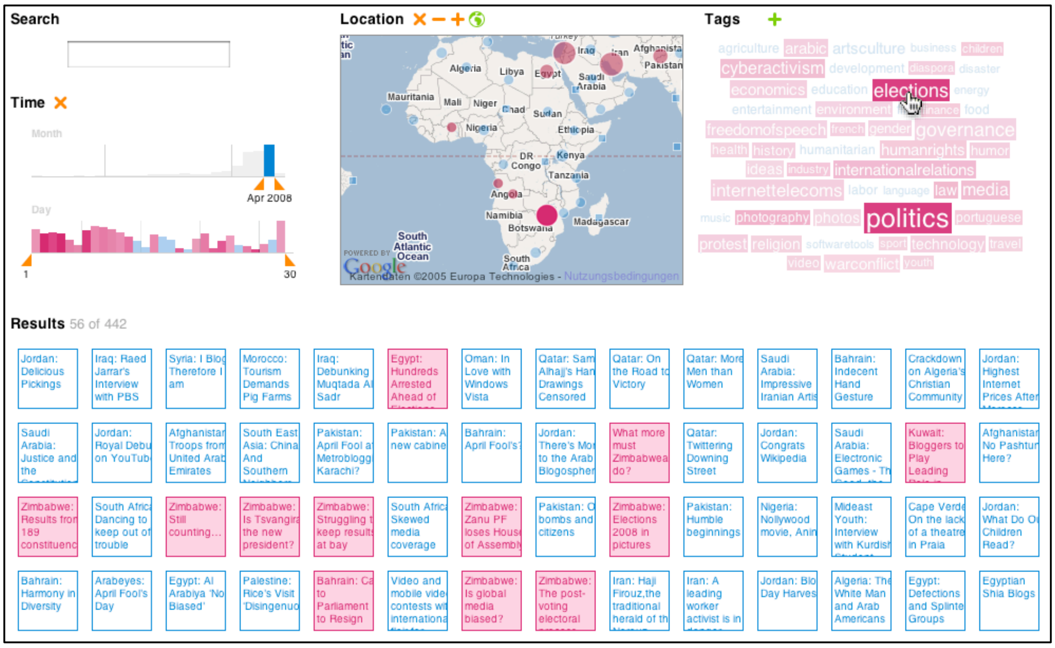


Figure 4.8: Brushing with three linked VisGets.

The degree of relatedness between visual elements in multiple VisGets usually differs, since each element can represent different quantities of collection items. Instead of having a binary type of linking and brushing, where weakly related elements are highlighted as much as strongly related elements, weighted brushing represents varying degrees of relatedness. The highlighting of linked visual elements is based on how much association there is between information items and the currently brushed element. For example, the currently brushed or activated element A in Figure 4.9 is highly related to the visual element B, as they both represent the same information items, possibly in different VisGets. The visual element C is weakly related to A, as it shares only one associated information item with A. The visual element D has no relation with A, and is therefore displayed with the default color.

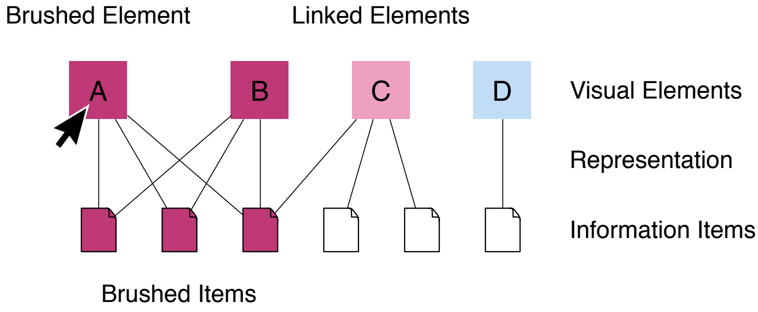


Figure 4.9: Weighted linking and brushing.

Information items can assume either one value or several values within a particular dimension. For example, an information resource usually has one publication date, while it can have multiple topics or tags associated with it. This has implications for the design of linking and brushing behavior within an activated VisGet. If the information seeker, for example, hovers over a tag, other tags in the same VisGet might be correlated and thus be highlighted. This provides the information seeker with visual cues indicating relations within the same dimension, in addition to highlights in other VisGets.

4.4.2 Query Refinement

An individual VisGet is used to constrain the query bounds of a particular information dimension through visual elements and controls that can be directly manipulated using the mouse pointer. In the implemented VIEW system, it is possible to use several VisGets for visual query formulation along multiple dimensions. By constraining the query bounds of multiple VisGets the query becomes more refined. Every refinement of a query triggers changes in the result list that are also reflected in all linked VisGets through animated transitions. Query parameters can be set in multiple dimensions, providing a technique for combining multi-dimensional parameters into a logical conjunction. In the following two examples are given.

Example 1 Consider an interest regarding politics on the Caribbean Islands around the middle of February 2008. Figure 4.10 shows steps using VisGets to follow this vague information need. A first step could be the selection of the circle close to the Caribbean on the geographic map, which highlights, in pink, the information items in the results list that are associated with the location of interest. In response to this action, the location VisGet zooms in to show the region of the Caribbean in more detail and the other VisGets update accordingly. Then one of the larger tags 'politics' could be selected from among the tags in the tag VisGet. Next, the temporal VisGet could be used to drill-down to the month of February. Brushing some of the days around the middle of the month of February highlights several items about Fidel Castro's retirement. Hovering over one of the result items shows the detail overlay of the news item "Castro Steps Down". Note that during the flow of refinement the results list becomes shorter and more refined.

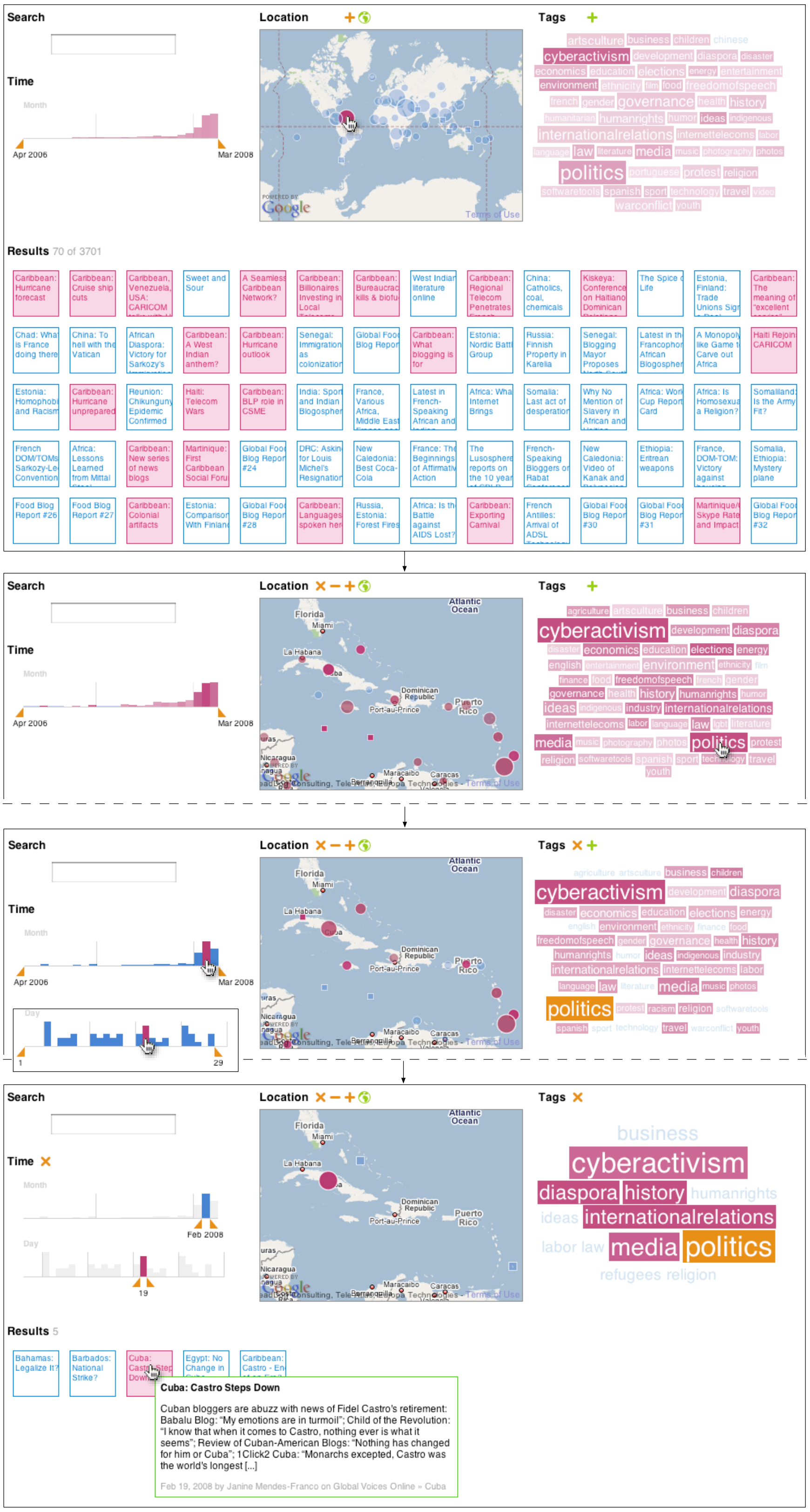


Figure 4.10: Example 1: Query refinement through location, tags, and time VisGets.

Example 2 It is also possible to begin with a conventional text-based search query, and then refine it using the VisGets. One could, for example, be interested in recent environmental topics mentioning storms (see Figure 4.11). At first 'storm' could be entered into the textual query box, after which all VisGets are updated visualizing only posts that contain the word 'storm'. For example, the tag VisGet would show that posts containing 'storms' are predominantly about the environment. Selecting this tag would disambiguate the query from posts that do not use the word 'storm' in connection with the environment. After that, the query could be refined further along the temporal dimension with the time VisGet to select only the last three months. Few items remain and the location VisGet indicates that a storm may have recently caused discussion about a place southeast of Africa. Selecting the circle on the map narrows down the results to two items, both indicating a storm that hit Madagascar. Hovering over one of the items shows a graphic displaying the severity and extent of the storm.

4.5 VisGets within the Web Browser

Coordinated VisGets constitute key elements of the implemented visual information exploration system that can be accessed through a Web browser. The system utilizes conventional Web browser features such as the address bar, page title, browser history, and bookmarking (see Figure 4.12).

The Web address bar showing the URL contains all relevant parameters of the current selection. As the information seeker changes VisGet parameters the Web address is updated immediately. The title of the Web browser window displays the current selection parameters in a more readable way. Only parameters from VisGets that have active query bounds are shown in the title.

In the text-based Web browser history menu, the titles of consecutive parameter changes are shown. The information seeker can use the Web browser history to jump to previous selections or undo an action using the back-button. Retrieved information can also be stored for later reference by saving the current selection as a bookmark. Similarly, it is possible to share findings with other people by sending the address, for example, via an instant messaging client.

4.6 Summary

This chapter described how VisGets work as part of an implemented VIEW system. In the following, the design will be summarized in terms of the design goals stated in the previous chapter (see Section 3.3):

1. **Enable casual formulation of complex queries**

   Queries involving multiple dimensions can be formulated visually using several VisGets. Conceptual dimensions that are difficult to specify in a conventional search

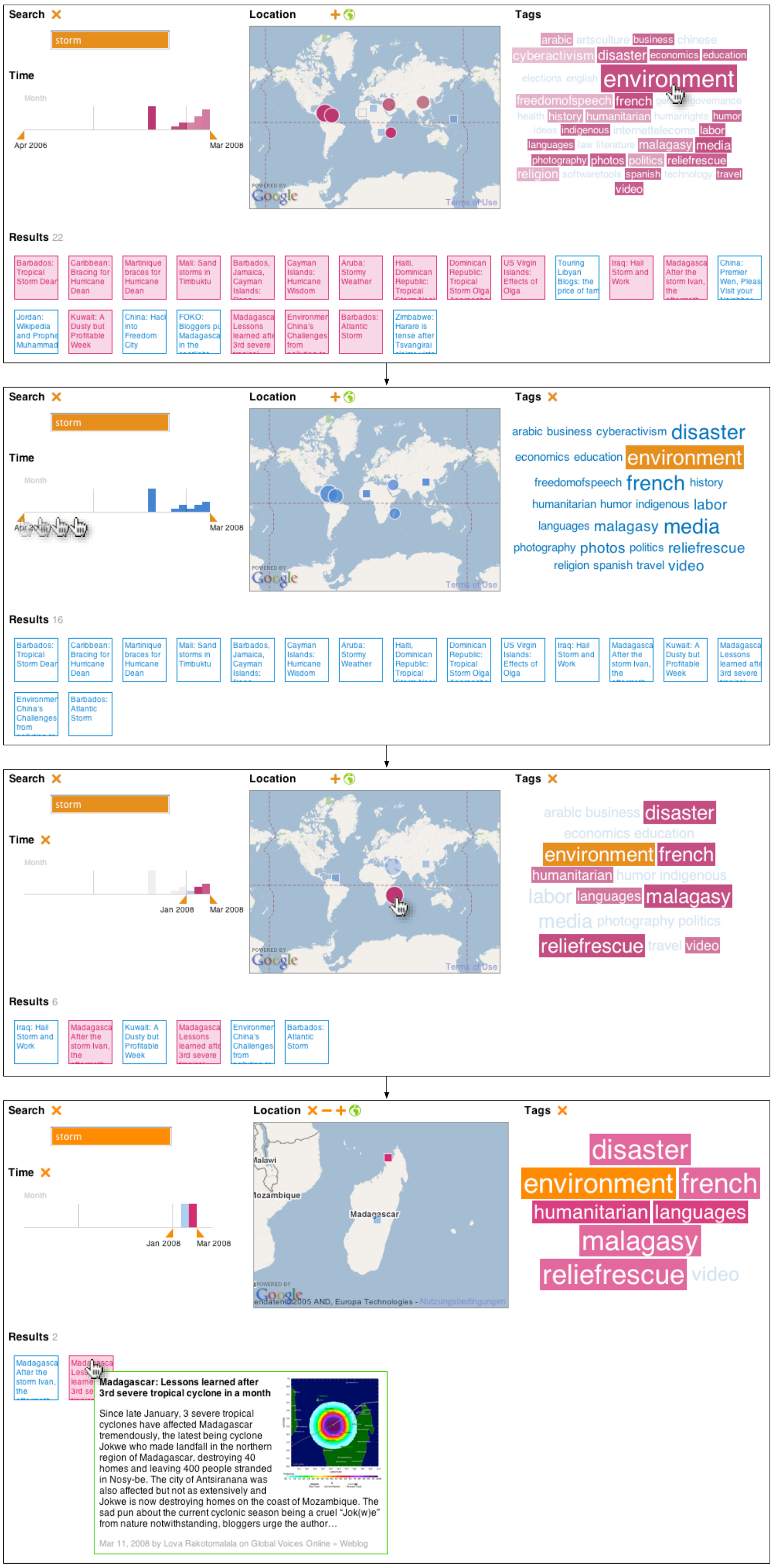


Figure 4.11: Example 2: VisGets complementing a conventional search query.

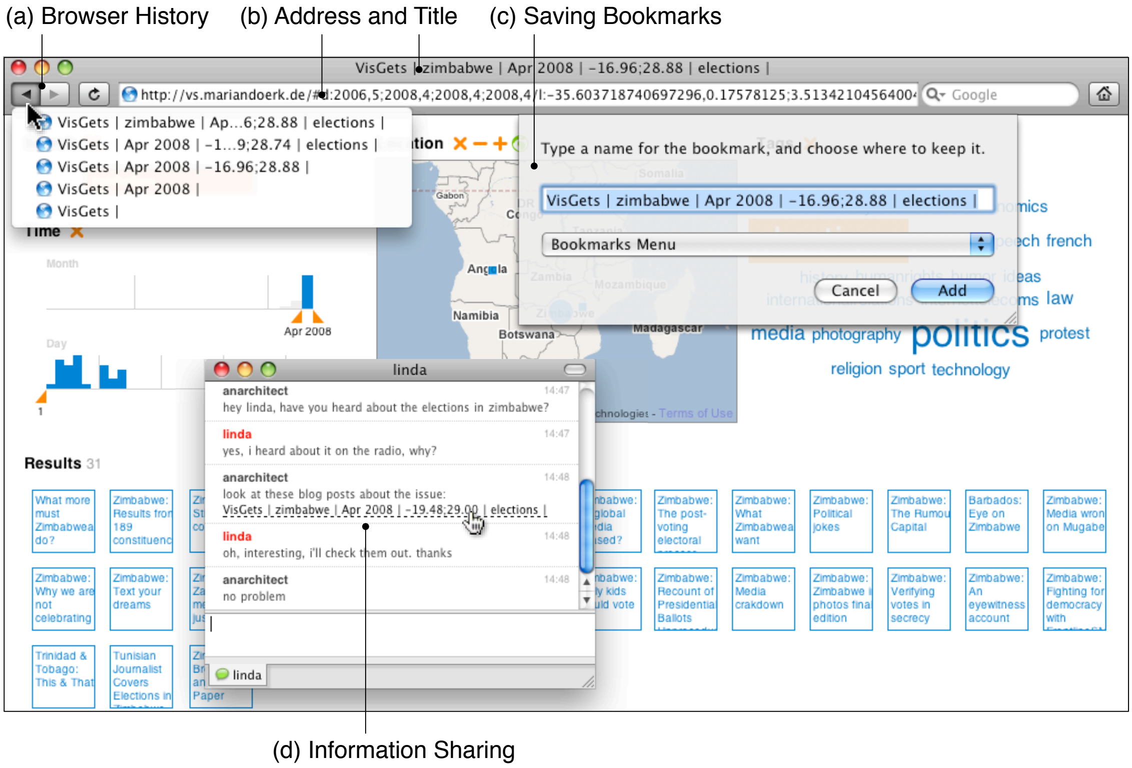


Figure 4.12: The VIEW interface utilizes conventional Web browser functions: (a) undo functionality and interaction history are available via the Web browser history, (b) the address field and page title reflect the current set of VisGet parameters, (c) findings can be stored using the bookmark function of the Web browser or (d) shared by sending the address using, for example, an instant messaging application.

query such as time or location can be set by interacting with a VisGet utilizing the mouse pointer.

2. **Summarize information collections visually**

   Every VisGet depicts overviews of a particular conceptual dimension. Interrelations between query dimensions can be explored by means of weighted brushing. Selected, activated, and filtered items are indicated using a consistent color scheme.

3. **Visualize bounds of query dimensions**

   The currently selected query parameters of a VisGet are indicated by orange controls and the colors used for visual representations of information sources. Query bounds can be modified by interacting with these elements. Switching between the selected range and the unfiltered view can be done by a reset control that is part of every VisGet.

4. **Visualize query changes**

   As a query parameter is changed in a VisGet, the whole interface is updated by way of coordinated query refinement. Animated transitions are employed where possible.

5. **Use integrated dynamic manipulation**

   Every VisGet includes interactive controls embedded into the visualization that allow the direct manipulation of the query parameters using the mouse pointer. The types of user interactions include clicking, dragging, and hovering.

6. **Provide information drill-down**

   Constraining a query dimension using the corresponding VisGet reveals more detail in the particular dimensions and at the same time refines the result list. Hovering over individual results provides more detailed information about the information source.

7. **Provide interaction history**

   The navigation history of the Web browser lists previous interaction steps and allows the information seeker to switch between them. A particular step along an exploration process can be saved as a bookmark.

8. **Enable lightweight information sharing**

   All query parameters are included in the URL displayed in the address bar of the Web browser. Sharing a selection of information sources requires only sending this address, for example, via email or instant messaging.

As the appearance and functioning of the VisGets have been explained, the next chapter discusses the implementation of the VIEW system.

5 Implementation of a VIEW System

I have implemented a Web-based system that follows my VIEW approach (Chapter 3) to better support visual exploration of large information collections utilizing VisGets (Chapter 4). This chapter discusses the challenges and decisions related to the realization of the system. The general overview of the VIEW system is described in Section 5.1 by explaining its software architecture, implementation structure, and runtime behavior. The challenges of data processing (Section 5.2), reducing bandwidth use (Section 5.3), implementing Web-based visualizations (Section 5.4), and converting query parameters (Section 5.5) are discussed in greater detail in this chapter.

5.1 Overview

For the implemented VIEW prototype, the basic visualization pipeline can be applied, with a particular role allocation between Web server and Web browser (see Figure 5.1). The earlier, more data-centric steps are assigned to the Web server, while the Web browser is responsible for the later, more visualization-specific steps. This architectural separation has been made due to the data processing capabilities of server-side databases and the interaction facilities of the Web browser.

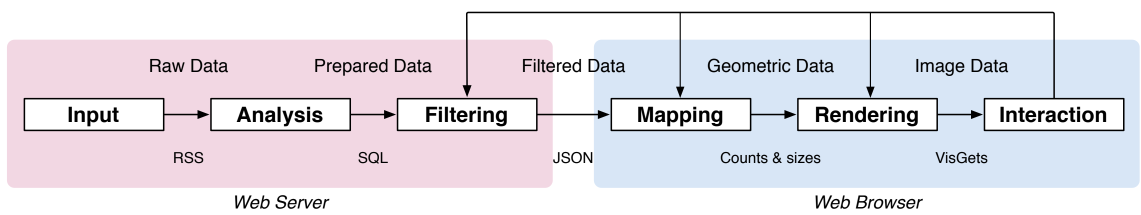


Figure 5.1: Visualization pipeline applied to the Web.

5.1.1 Architecture

The realized software spans two contexts: data-centric functionality such as analysis and filtering is implemented on the Web server, and visualization-specific logic such as presentation and interaction is implemented in the Web browser (see Figure 5.2).

Web Server On the Web server, aggregation of multiple RSS feeds, data analysis, and filtering is done. PHP was chosen as the server-side programming language, since it is the most widely deployed and established Web development language that is available in both low-cost and high-end server setups. CakePHP [20] is used as the PHP programming framework that supports the Model-View-Controller paradigm and allows for convenient and flexible database operations. Upon HTTP requests from the Web browser, the Web server provides filtered data in the JSON (JavaScript Object Notation) format as HTTP responses. MySQL is used as the relational database for storing processed data from RSS feeds.

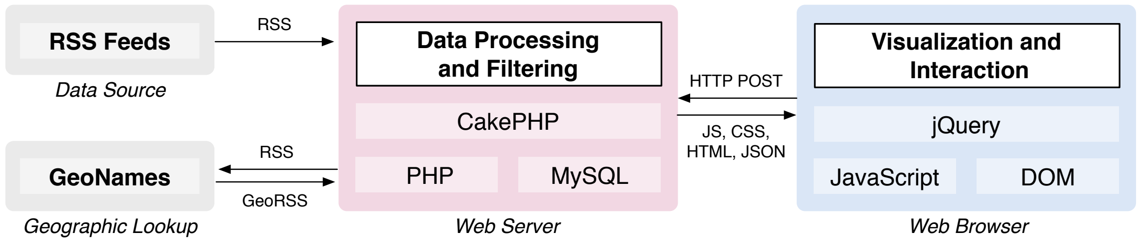


Figure 5.2: Architecture of the implemented VIEW system.

Web Browser The visualization and interaction logic of the VIEW system is implemented in JavaScript, which can be executed by most current Web browsers. Client-side programming code, the general interface, and the style descriptions are initially transmitted as JavaScript, HTML, and CSS files once the system is accessed using a Web browser. The actual processed and filtered information sources are retrieved by the browser as JSON data that are mapped and displayed by VisGets and result list. jQuery [53] has been used as a JavaScript framework that abstracts away the differences between multiple Web browsers. jQuery was chosen, because it enables both sophisticated query operations on the DOM (Document Object Model) as well as asynchronous communication with the server via the XMLHttpRequest object.

In addition to the Web server and Web client, the architecture of the implemented VIEW system includes multiple, distributed RSS feeds as data sources that can be added and removed through the Web browser, and the GeoNames [36] Web service that provides functionality for geographic lookup [104].

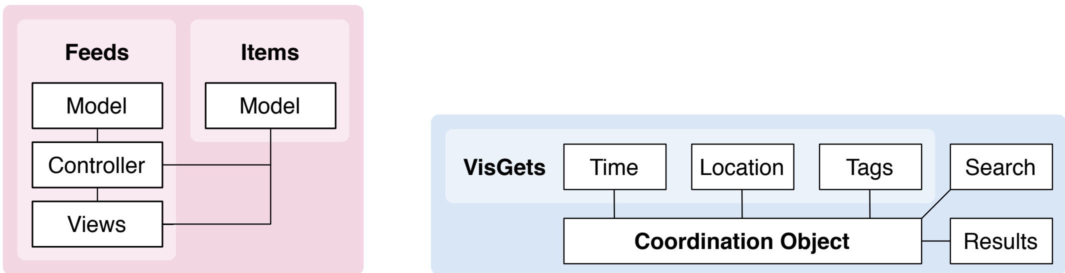
5.1.2 Class Composition

To outline the scope of the realized software, the major implemented classes on the Web server and Web browser are now briefly described.

Server-side

The server-side implementation is divided into two groups of classes (see Figure 5.3(a)). Aggregation of RSS feeds is done by the *feeds* part, while *items* are responsible for processing individual information items:

- **Feeds**
  - Model: The feed model handles retrieval and aggregation of feeds from their Web sources. New feeds are added to the database and existing ones are updated.
  - Controller: The feed controller is responsible for handling HTTP requests by calling model methods, passing parameters, and returning views.
  - View: Feed views are those files that are sent back to the Web browser, such as generated HTML or JSON.
- **Items**
  - Model: The item model processes information items by extracting multiple dimensions and saving items to the database. Furthermore, SQL queries are assembled for filtering information items.
  - For items, there are no separate views or controllers, since functionality for JSON generation and HTTP handling is provided by the feed views and controller.



(a) Web server.
(b) Web browser.

Figure 5.3: Main classes of the VIEW software.

Client-side

The part of the VIEW system that is running in the Web browser implements interaction and visualization functionality. As depicted in Figure 5.3(b), the client-side implementation comprises a coordination object and classes for VisGets, search query, and results:

- **VisGets**

  Every VisGet has its own class, in which all functionality specific to this VisGet and its dimension is implemented. This includes mechanisms for parsing and transforming query parameters and the logic for interaction and visualization.

- **Search and Results**

  Similar to the VisGets, the search query and the result list are realized in separate classes, which handle their specific events and DOM manipulation.

- **Coordination Object**

  To keep coordinated interaction as simple as possible, VisGets and other interface elements are not directly linked to each other. Instead, a coordination object pools all interaction, so that a user event triggered in one interface element is linked with the remaining interface. The coordination object implements all client-side functionality that is not specific to a particular VisGet or other interface element. This includes combining query parameters of the individual VisGets to issue data requests to the Web server.

5.1.3 Sequence of Processes

The runtime behavior of the implemented system can be described by means of four interconnected runtime scenarios: *initial access*, *query change*, *linked brushing*, and *feed update*. In the following, the sequences of processes for each phase are explained by using the aforementioned classes.

Initial Access

This phase describes how the VIEW interface is launched and how the VisGets are initialized (see Figure 5.4).

1. The Web address of the VIEW system is accessed to load the interface. The address may include query parameters.
2. The feeds view returns the files necessary for the interface. The coordination object is initialized, which in turn initializes all interface elements: VisGets, search box, and results.
3. If the Web address includes parameters, these are parsed. Otherwise default parameters are set. An initial data request is sent via HTTP to the feeds controller.
4. The controller passes the parameters to the models that transform the parameters into SQL statements and thus retrieve the desired information sources from the database.
5. The controller generates a JSON view that is sent back as an HTTP response to the coordination object running in the Web browser.
6. The data is loaded by all VisGets, which map the data to the visual elements that will be added to the DOM. The result list displays the items.

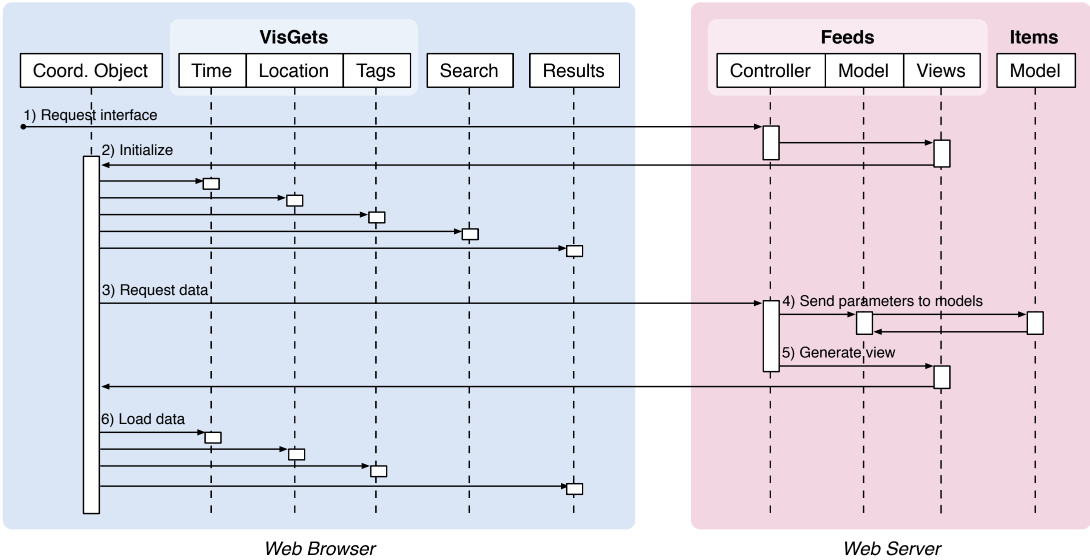


Figure 5.4: Sequence of processes during the *initial access* phase.

Linked Brushing

Hovering over visual elements in one VisGet highlights related items in all VisGets and in the result list. This coordinated highlighting is realized only within the client-side part of the VIEW system (see Figure 5.5).

1. Moving the mouse pointer over a visual element, for example, a circle on the map of the location VisGet, prompts the VisGet to determine all information item ids that are represented by this visual element and pass them to the coordination object.
2. The coordination object propagates the list of highlighted information items to the VisGets and the result list, which highlights the corresponding visual elements by degree of relatedness (see Section 4.4.1).
3. Once the mouse pointer leaves the visual element, brushing is terminated.
4. The coordination object prompts the VisGets and results to remove all highlights.

Query Change

A query change can be initiated by any VisGet, the search box, or by using the back button of the Web browser and thus changing the Web address. In the following, an interaction with the time VisGet is used as an example (see Figure 5.6).

1. The query change is initiated through the time VisGet, which transforms, for example, changed positions of the sliders into new bounds of the temporal dimension. Furthermore, the time VisGet determines which items remain within the bounds and which are to be filtered out.

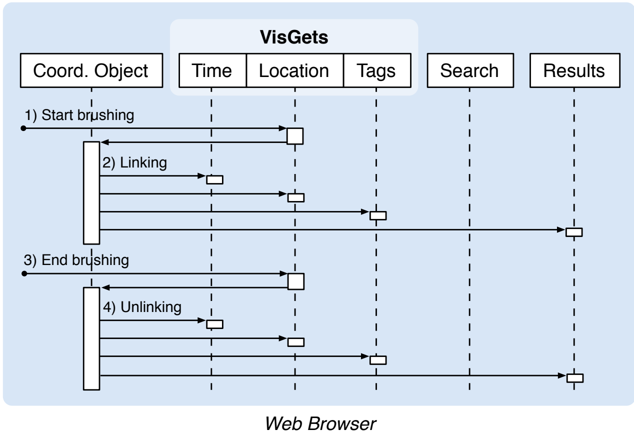


Figure 5.5: Sequence of processes during the *linked brushing* phase.

2. The query parameters including the new temporal parameters are sent as an HTTP request to the server to get additional information items. Only those items that are not currently loaded in the Web browser are retrieved. In Section 5.3, this mechanism will be discussed as Delta Queries.

3. By knowing which items remain and which are to be deleted, VisGets and the result list can release the items that do not meet the current query parameters, before additional information items are returned by the Web server.

4. The VisGet and result objects load the newly retrieved items, then process and map them to visual representations and DOM elements.

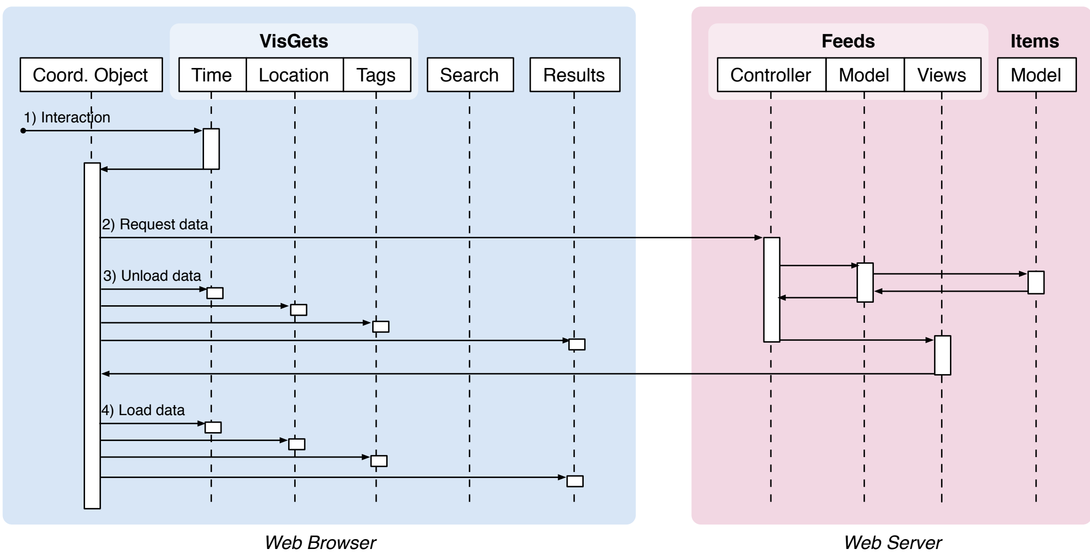


Figure 5.6: Sequence of processes during the *query change* phase.

Feed Update

Feed updates are periodically initiated from the Web browser, to keep the information space current. Updating is transparent for the information seeker, in the sense that it does not require user interaction or attention (see Figure 5.7).

1. The coordination object periodically sends HTTP requests to the server-side feed controller. The feed with the oldest modification date is selected to be updated.
2. The model requests a current version of the feed and examines which items of the feed are to be added by the item model.
3. The item model processes the individual feed items, extracts information, and saves the new item in the database.
4. Next time a query change is initiated, the new information items will be retrieved from the Web server and appear in the interface (see *query change* phase).

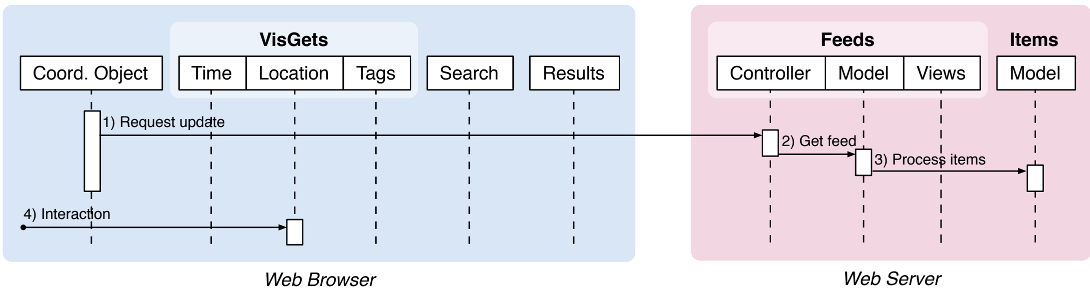


Figure 5.7: Sequence of processes during the *feed update* phase.

5.2 Data Aggregation and Integration

In the implemented system, information items are extracted and constantly updated from RSS feeds, which are added through the Web interface. Extraction of title, description, and date information from RSS feeds is straightforward, and is done using the MagpieRSS library [65]. However, extracting location and category information is more difficult, since this information is not always presented in a consistent format.

Tags can be represented in RSS feeds in different ways. The native approach in RSS is to put tags into `<category>` elements that are sub-elements of the `<item>` element. Tags can also be included as part of an extension to RSS, such as MediaRSS [111] that is used by the photo-sharing site Flickr. Furthermore, blog posts are often tagged using links to blog aggregation sites, such as Technorati [97], with the link's `rel` attribute set to "tag". In RSS feeds, these tag links appear in the description, and have to be parsed; the implemented system utilizes regular expressions for this.

While the standard specifications for geospatial information embedded in RSS are unambiguous, the number of RSS feeds that include latitude and longitude on a per-item basis is limited. At least, often titles or descriptions include geographic indications, such as city or country names. To add explicit geographic information to RSS feeds that include geographic names, the GeoNames Web service was used, which (among many other gazetteer functions) extends regular RSS feeds into GeoRSS feeds. The Web service takes the URL of a non-geospatial RSS feed as an input, then returns the same feed with pairs of longitude and latitude added to items, if sufficient geographic indications are present.

Descriptions included in RSS feeds can vary greatly in terms of length and formatting. To provide a consistent display of information in the interface, any markup other than formatting with italics and bold is excluded and the length of the description is limited. If one or more images have been added to an item, the source URL of the first image is parsed using a regular expression. This image will then be displayed in the detailed views of individual information items.

5.3 Delta Queries

Responsiveness is a vital feature in interactive information visualization. In the Web context, this requirement places practical constraints on the number of client-server transactions and the data volumes exchanged, because of round-trip latency and network bandwidth consumption. As a result, one of the key design considerations was avoiding unnecessary data movement, especially for redundant data.

To provide fast and comprehensible updates, I developed *Delta Queries*. After a visual query parameter has been modified, the corresponding VisGet determines which items are to be removed, which items remain, and what kind of overlap exists between the current selection s_i and the new selection s_{i+1} of information items. Resulting change in the information set is referred to as the delta.

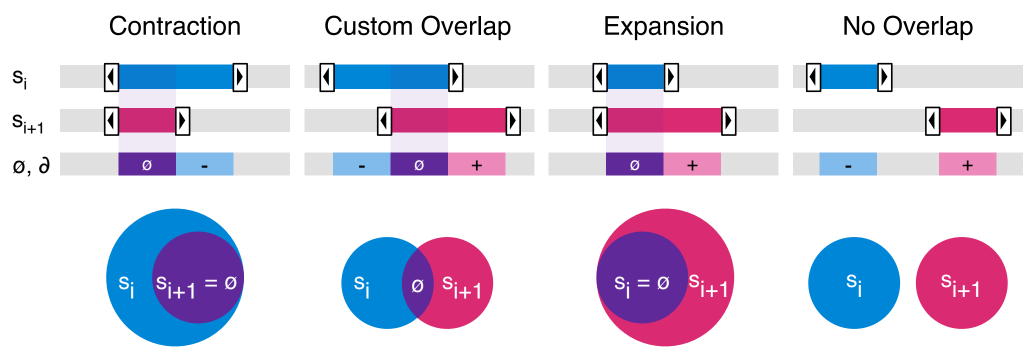


Figure 5.8: Types of overlap depicted by means of two consecutive selections s_i, differences δ, and overlaps $ø$.

Four types of overlap have been identified, which are depicted with sliders and Euler diagrams in Figure 5.8. By determining the overlap (if any), it becomes possible to reduce the number of items requested from the Web server, reducing network latency and thus the time to update the interface. Furthermore, animated transitions are provided to indicate removal or addition of information items in the result list, as well as the changes happening within the VisGets.

After several query changes the VIEW interface should show the same information items as if it was opened anew with the same query parameters. To achieve this, the implementation of Delta Queries requires careful tracking of information items. Once the VisGet determines the ids of the information items that do not comply to the new query anymore, these items are removed from all VisGets and the result list. Depending on the VisGet, animations may convey interface updates. Ids of the retained information items are sent within the HTTP POST request to the Web server, indicating they can be excluded from the returned JSON data. In the special case of a contraction overlap, the query to the Web server can be omitted altogether.

5.4 Visualization and Interaction

Even though the Web was not conceived for advanced computer graphics, it is possible to realize simple interactive Web-based visualizations that are effective and appropriate by making use of current technological advancements. The presentation and interaction functionality of VisGets is realized by means of JavaScript and CSS. In particular the methods for accessing and altering DOM elements and applying events and animations utilize the jQuery library. This section explains how each VisGet object computes and draws visual representations and handles interactivity.

5.4.1 Time VisGet

The time VisGet features interactive bar charts and sliders that are implemented as `<div>` elements with styling applied for size and position. The size of an individual bar—representing a month or a day—is determined by the relative amount of information items associated with this month or day. Once the amount of items per day or month changes, the size of each bar is changed by an animated transition.

Bounds of the temporal dimension can be changed using the mouse pointer either by clicking an individual day or month bar, dragging a slider, or resetting the VisGet. For each case, events are associated with the corresponding DOM elements that will prompt the time VisGet to assess what kind of overlap between old and new temporal bounds exist, and consequently determine elements that have to be deleted. Their ids and the type of overlap are passed back to the coordination object.

The time VisGet implements weighted brushing utilizing the class and style attributes of the DOM elements that represent days and months. To those DOM elements that are highlighted, the class `active` is attached which is defined in the CSS file with a different background color. Furthermore, opacity conveys a relative degree of relatedness. Visual elements that represent more of the brushed information items are associated with a larger value for the opacity attribute than those that represent fewer or none of the brushed information items.

5.4.2 Location VisGet

Visual representation and interactivity of the location VisGet is implemented utilizing the Google Maps API [41]. The visual elements, circles and squares, are realized as map markers that can have different sizes and images. A square stands for one individual information item, whereas a circle represents multiple information items that are either at the same location or are merged into a region cluster. The implemented clustering algorithm can be regarded as simplified agglomerative hierarchical clustering: all points are successively considered; if the distance between the current point and the closest cluster is within a given threshold, the point is added to the cluster and the cluster center is computed anew; if there is no cluster close enough, the point itself becomes a new cluster. The size of a circle is determined by the number of information items that are associated with the cluster.

Spatial query parameters are represented by the bounds of the map, which can be changed by dragging the map, zooming in or out, and clicking the circles or squares. Interaction with the geographic map itself is not implemented by the location VisGet, since it is already provided by the Google Maps API. If the bounds of the map changes, the location VisGet determines the new spatial query parameters and recalculates which items are to be removed and which remain.

To display the locations as map markers, the circle and square graphics have to be prerendered images. During weighted brushing, the corresponding highlighted squares and circles have to be loaded as image files, as well. Therefore, the location VisGet swaps the marker images depending whether a location includes brushed information items. Different degrees of relatedness are realized by using several images with differing opacities.

5.4.3 Tag VisGet

The tag cloud is realized using text elements with differing sizes that are specified through the style attribute. Depending on the number of information items associated with a tag, the tag is displayed with font size ranging between a minimum and a maximum. When query parameters are changed, the sizes of the fonts are changed by animated transitions.

The tag VisGet allows the specification of topical query parameters as bounds for the tag dimension. Unique identifiers and events for clicking a tag are associated with every textual element of the tag cloud. Clicking a tag activates the tag as a query parameter and the tag VisGet will compile the necessary parameters—type of overlap, remaining items, and items that are to be removed—and propagate the query change to the coordination object. Clicking a tag a second time will deactivate it as a query parameter and initiate the corresponding query change.

The weighted highlights for linked brushing are implemented similarly to the time VisGet: a class for linked visual elements that represent brushed information items is attached to tags and the tag opacity is adjusted according to the level of relatedness, i. e., the relative number of brushed items.

5.5 Query Parameter Conversion

The query parameters determine which information items are visualized in the VisGets and are displayed in the result list. In the implemented system there are four types of dimensions—time, location, tags, and text search terms—whose bounds represent query parameters that can be altered utilizing the mouse pointer through VisGets and in the case of the search terms by means of a conventional text field. In the implemented VIEW system, query parameters have to be transformed between four different representations (see Figure 5.9):

- URL-encoded parameters for HTTP requests and the Web address,
- SQL-encoded parameters as part of database queries,
- readable query parameters for browser title, and
- visual query parameters as part of VisGets.

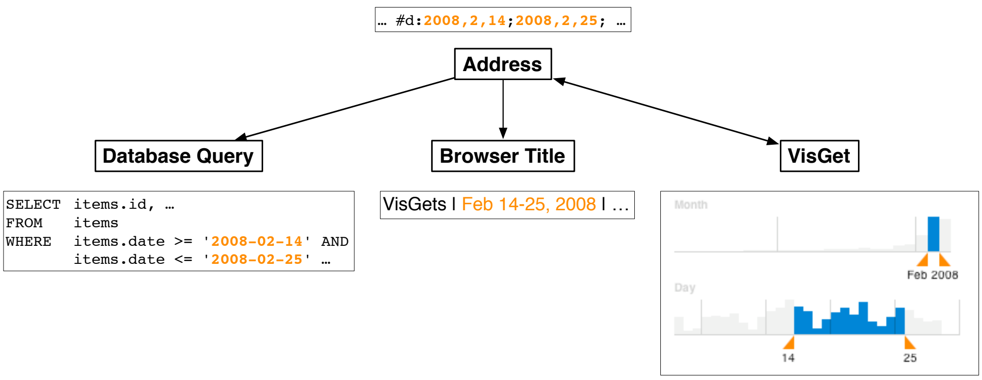


Figure 5.9: Query parameter conversion, here shown with temporal dimension.

The challenge is to represent query parameters in these different forms, and apply transformations between them without sacrificing accuracy or semantics.

5.5.1 URL-encoded Parameters

In the implemented system, every query state is encoded as a URL. This allows for saving parameters as bookmarks, sending these to other people, and utilizing the Web browser history as an interaction history. Representing query parameters in URLs is also essential in the distributed, Web-based environment, in which the VIEW system has data-centric logic running on the server and presentation logic within the browser.

Temporal, spatial, and topical constraints are transformed into URL-encoded representations, so that both the server and client are able to interpret and generate them. The URL-encoded representation of the query parameter is condensed to keep the address short yet readable (see Figure 5.10). Individual query dimensions are separated by slashes ('/'), one dimension's query parameter is introduced by its first letter and a colon (':'), and within a query dimension, sub-parameters are divided by commas (',') and semi-colons (';').

```
…#search/time/location/tags

…#search:query
/date:year,month;year,month
/location:lat,long;lat,long
/tags:tag1;tags2;…
/order:time

…#s:foo/d:2007,5;2007,5/l:0.123,-0.345;0.123,-0.345/t:bar
```

Figure 5.10: Parameter representation in URL.

Because JavaScript code executed by the Web browser can only change the local part of addresses, query parameters for the address in the Web browser are put after the hash sign ('#'). This allows the client-side code to change query parameters in the URL, once the information seeker has modified them using a VisGet.

5.5.2 Parameters in SQL Statements

To retrieve the information items fulfilling the current query from the MySQL database, the URL-encoded parameters are transformed into SQL statements. Parameters are converted into `WHERE` constraints, one for every query dimension.

The temporal constraint is an interval between two dates either along the month or the day range. For example, if the first three months of 2008 are selected, the temporal constraint for the `WHERE` part of the SQL statement is formulated as follows:

```
items.date >= '2008-01-01 00:00:00' AND
items.date <= '2008-03-31 23:59:59'
```

The spatial query parameter is defined by the most northeast and southwest points on the map that are utilized as intervals for latitude and longitude. To represent spatial constraints around the longitudinal discontinuities at +180° eastward and −180° westward, more complex `WHERE` constructs are generated than the following example, where the map is zoomed in to Calgary and Banff:

```
items.lat > '49.3' AND items.lat < '52.6' AND
items.lng > '-117.6' AND items.lng < '-110.5'
```

The tags are stored in an indexed MySQL row, which allows the use of `LIKE` comparisons. The `WHERE` condition for an activated 'environment' tag looks like this:

```
items.tags LIKE '%environment%'
```

The search parameter also relies on indexed fields. The search query is applied to the title, description, and tags fields. The fields are compared with the search query using the `MATCH` operator. Multiple search terms are transformed into a logical conjunction. Searching for 'world peace' would translate to this `WHERE` condition:

```
MATCH (title,description,tags)
AGAINST (' +world* +peace* ' IN BOOLEAN MODE)
```

Depending on which VisGets are active, only a subset of these query dimensions may be added to the `WHERE` condition of the SQL query.

5.5.3 Title Parameters

To provide useful Web browser titles, history entries, and bookmarking function, query parameters have to be transformed into textual representations that are readable and allow the distinction between query changes. While this is uncomplicated for tag and search parameters, it is more difficult for the time and location VisGets. Dates of temporal constraints are transformed from a purely numerical representation to a conventional form, where the month is written as a word. Depending on the type of temporal selection, the range can be stated as, for example, "May 14, 2008" or "May-August 2008." For the location VisGet, the rounded, geographic tuple of the map center is used. However, in combination with the online GeoNames gazetteer this could be improved by using the proper names of geographic entities visible on the map, such as continents, countries, and cities.

5.5.4 VisGet Parameters

As discussed in Section 5.4, query bounds are implemented in different ways in the visual representations of the VisGets. Visual query parameters can be slider positions, map bounds, and tag selections:

- The temporal query bounds are represented by the position of triangle-shaped `<div>` elements used as sliders. The pixel positions of these sliders can be retrieved through the browser's DOM and set through the style attribute.

- The spatial query parameter is represented by the map's bounds, which are defined by the northeast and southwest corners. These points can be easily retrieved and set using the functions provided by the Google Maps API.
- The selections of one or multiple tags as filters or query parameters is set using the class parameter of the respective textual element that shows the tag.

5.6 Summary

This chapter explained the challenges and decisions concerning the implementation of a VIEW system. First an overview of the system was given by explaining its Web-based architecture, discussing the components of the implementation, and outlining its runtime behavior by means of four interconnected phases. Then the challenges of aggregating and integrating distributed and diverse RSS feeds were discussed, in particular, with regard to the spatial and topical attributes. Delta Queries was presented as a technique to decrease bandwidth usage, when two successive selections overlap. Subsequently the realization of Web-based visualizations based on JavaScript and CSS utilizing the Google Maps API and the jQuery library was discussed. And finally, query parameter conversion between multiple representations was described as it has been engineered.

The implemented software represents an initial attempt to follow the VIEW concept and realize the VisGet design. The forthcoming chapter explains how an exploratory evaluation has been undertaken to make a preliminary assessment of the perceived usefulness, potential uses, and problems of the implemented artifact and the underlying ideas.

6 Exploratory Evaluation

Throughout development of concept, design and realization of the VIEW approach, several types of open-ended evaluation methods have been undertaken to generate ideas, estimate usefulness, and uncover problems:

- *Brainstorming.* During the early stages of this work, multiple brainstorming sessions with colleagues and advisors took place to discuss the idea, related research, preliminary design, and potential of visualization for information seeking.
- *Observational user study.* An initial user study with 10 participants has been undertaken with the goal to learn about how information seekers would perceive a realized VIEW system in regard to its usefulness, to uncover conceptual drawbacks, and to reveal implementation problems.
- *Focus group.* In an informal and voluntary follow-up meeting a focus group session was conducted. In this focus group, four study participants shared their ideas about how visual information exploration needs to be improved and applied to additional domains.

The results of these evaluation methods guided design and implementation of the VIEW system. My research interest was mainly focused on the perceived usefulness, possible applications, and generation of further ideas. In this sense, the implemented VIEW system can be understood as a tentative prototype comparable to an interactive sketch [42] and its evaluation as an attempt to spark discussion and open up new perspectives for information seeking and visualization.

In this chapter, relevant discourse on evaluation of information visualization is outlined, the conducted observational user study and its results are discussed, and finally the procedure and outcomes of the focus group are explained.

6.1 Evaluating Visualization Systems

In InfoVis and HCI literature, there have been discussions about the quality of visualization evaluation [7, 27] and whether early usability studies are suitable at all [42]. Evaluating visualization tools has been identified as particularly challenging as they often support information seekers by "answering questions you didn't know you had" [80].

Researchers have discussed what kind of evaluation techniques would be most suitable to study people at seeking and analyzing information through visualization and search systems. While benchmark tests facilitated studying low-level perception and specific tasks, it has been argued that an open-ended task design is more appropriate for examining more high-level cognitive effects [75]. Following this assumption, assessing the overall usefulness of a tool demands a wider range of evaluation methods and metrics [57, 84]. While search systems traditionally have been evaluated with precision and recall, it was stated that evaluation of exploratory search systems should put more emphasis on information seekers and their tasks, which would yield better feedback on the adequacy of a system [58]. For studying the usefulness of visualization tools, focus groups can be an appropriate way to obtain rich data and new ideas [69].

6.2 Observational User Study

After a VIEW system was implemented (see Chapter 5), an initial user study with ten participants was undertaken in the Interactions Lab at the University of Calgary. The purpose of the study was to see how the system was used and perceived by information seekers in terms of its viability for information exploration, and to expose problems with its design and realization.

6.2.1 Participants

Participants, 4 female and 6 male, ranging in age from 19 to 37, were recruited through posters placed on bulletin boards across the university campus. Eight participants were enrolled in academic programs, such as computer science (3), civil engineering (2), economics (1), general studies (1), and bioscience (1). The remaining two participants were members of university staff. Participants self-estimated their Internet experience between 5 and 15 years and their Internet usage between 2 and 8 hours per day.

6.2.2 Setup and Design

Participants sat at a regular desk and used an Apple MacBook computer with a 20-inch external display, standard North American keyboard, and generic computer mouse. The computer was connected to the Internet, and the Web browser window was centered on the screen, which was set to a size of 1100 pixels wide and 980 pixels high. During the study, the data consisted of approximately 3000 articles extracted from several RSS feeds from the Global Voices site [37].

Participants were given study tasks printed on a sheet of paper, with spaces to fill in their written answers. Each study session lasted about one hour and the participants were remunerated. The researcher took notes during the study (e. g., of comments made by participants).

6.2.3 Procedure

At the beginning, each participant was given a tutorial by the experimenter explaining and showing how the system worked. Then the participant had free exploration time to become familiar with the VIEW system. After this, each participant was asked to perform two types of tasks.

The first set of tasks consisted of nine focused questions about current events and had specific countries or regions, topics, or dates as answers. Each question about a given dimension had clues from two additional dimensions. For example, one question was: "Which region declared its independence in February 2008?" Here the answer was spatial (Kosovo) and the clues were topical (independence) and temporal (February 2008).

The second set of tasks was more open-ended. Each participant was asked to imagine that he or she was a newspaper journalist, a health inspector for the World Health Organization, or a human rights investigator with Amnesty International. The tasks were to make fictitious travel plans corresponding to these roles, i. e., to uncover interesting news stories, track the global health situation, and monitor human rights issues.

After the tasks were completed, participants were asked to fill out a post-session questionnaire that consisted of Likert scale questions about the ease of making discoveries and seeing relationships using the VIEW system and the individual VisGets. After the participant had filled out the form, a semi-structured interview was conducted about perceived usefulness of the VIEW system developed in this work, problems during interaction with it, and possible improvements. The results of the questionnaire are presented below. The results from the semi-structured interview were used to better understand the results of the questionnaire; these are included in the discussion section.

6.2.4 Quantitative Results

The post-session questionnaire comprised ten five-level Likert scale questions on how well the interface and specific elements would support discovery of information, seeing relationships, gaining an overview, and access to detail information. Data from one participant had to be excluded due to an incompletely filled out form. The medians for all questions were 4, except for the following statement that had a median of 5: 'The time sliders allow you to easily discover interesting topics, events or news'. How often the levels of the scales from 'Strongly disagree' to 'Strongly agree' were chosen is shown in Figure 6.1.

6.2.5 Discussion

While the spatial and topical VisGets were rated similarly by participants of the user study, it was a bit surprising that the temporal VisGet was rated as particularly helpful for discovering information. In the observations and the semi-structured interview this result has been confirmed. During the study, it was noticeable that participants used the loca-

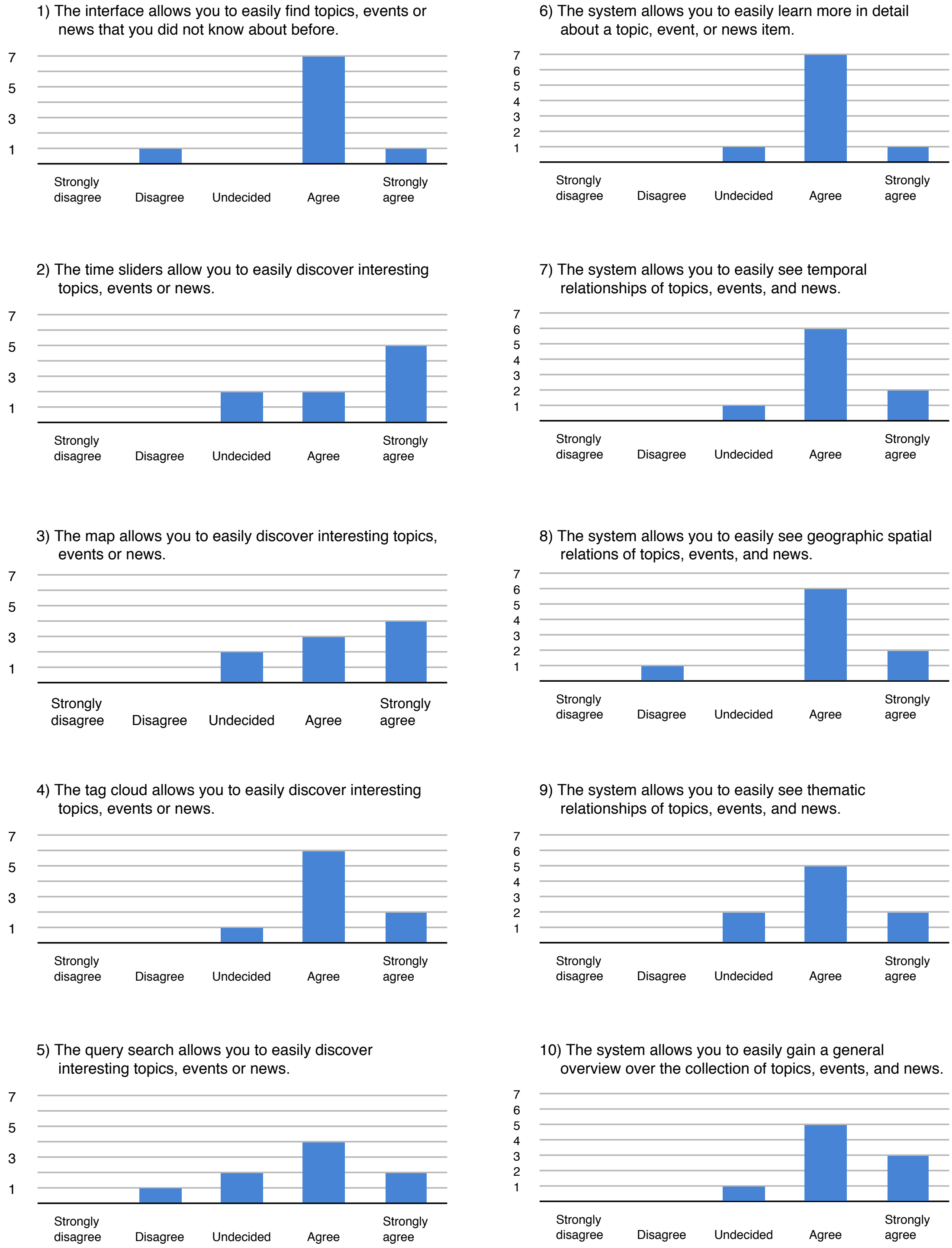


Figure 6.1: Levels of agreement stated in post-session questionnaire.

tion VisGet less than the time or tag VisGets for visual filtering. Spatial cues for countries were used by some participants as textual queries. In the interview, five out of ten participants chose the time VisGet as their preference for filtering through the information items, while four participants described the location VisGet as unnecessary.

When asked where such a visual search would be helpful, four participants mentioned news and current events, three noted possible value for academic research, and three regarded it as applicable to conventional search. When asked about suggestions for improvement, five participants mentioned that a more sophisticated sorting or ranking mechanism for the result list would be useful. Three participants mentioned 'next' and 'previous' buttons to browse through the results that could not be displayed. Furthermore, it was noted by four participants that VisGets would be more useful to have as an optional, complementary addition to conventional search.

These preliminary findings suggest VisGets can be of help for the information seeker in certain circumstances for specific concepts. The surprisingly positive opinion about the time VisGet may be due to the difficulty of expressing time in words as part of a textual query and the lack of temporal query building in current search systems. While the location VisGet was appreciated by most participants, having some participants strongly question its relevance for information exploration, suggests that it may not be as meaningful for representing information or useful for formulating queries as the time and tag VisGets. Even though map bounds may not be put into human-readable words, spatial concepts like country and city names were seen as sufficiently functional as textual queries. The degree to which location is meaningful for an information space, certainly depends on the type of information sources and warrants further investigation.

6.3 Focus Group

Study participants were invited to an informal focus group to discuss the research of Web-based visual information exploration and how VisGets could be extended in the future. Four interested participants from the user study came voluntarily and contributed their ideas, critiques, and thoughts about visual information exploration in a casual format. The session lasted about 75 minutes.

The following insights should be understood as subjective ideas from interested participants gathered in a discussion where the 'experimenter' engaged in the conversation. No claims about generality are made.

6.3.1 Present Web Use

At the beginning of the focus group an introduction round was held, in which participants briefly talked about the Web sites on which they spend most of their time online:

- Forums: with differing degree of participation.

- Webmail: Hotmail, Yahoo Mail, Google Mail, etc.
- Facebook: for news from other contacts, applications, embed videos, share photos.
- YouTube: watch videos, write comments.
- News: for example Google News.
- MySpace: find new music, look up artists, touring dates, new singles.
- Wikipedia: looking up specific things and browsing without particular target interests.
- Google: conventional Web search; also often instead of typing known addresses out of convenience, because Google is set as homepage.

It was noted by multiple participants that they usually use Google to search for specific YouTube videos, artists on MySpace, or Wikipedia articles instead of the local search functionalities of the respective Web sites. For example, searching for a music artist on MySpace was considered easier and faster with a Google search query such as "myspace artist name" instead of using the search tool of MySpace. Similar examples were mentioned for YouTube and Wikipedia.

6.3.2 Sketches

After the brief discussion of regularly visited Web sites, participants of the focus group were asked to draw a sketch of what they remembered of the system they had used during the study session a week earlier. The sketches resembled all major parts of the interface, indicating that these participants remembered the system well.

Later during the focus group, participants were asked to sketch visual information exploration interfaces that they would find useful for themselves. In addition to sketching suggestions for improvement, most of which were also raised during the user study, the sketches offered insight about how participants conceived the use of VisGets for Web sites they regularly visit, such as MySpace, Wikipedia, and online library catalogues. It was interesting to note how the general concepts of time, location, and tags were utilized for more specific dimensions that are meaningful for the different contexts. For the time VisGet, different meanings were stated, for example, the last time a profile page was updated or an article was changed, the publication date of a book, or the main time period of a subject. The location VisGet was also included in the sketches, for example, to be able to explore MySpace members by hometown or Wikipedia articles by the location of the main subject.

The ways how the tag VisGet was applied featured the most variation and innovation. Two participants embedded in their sketches the possibility of visualizing and exploring information through faceted tag clouds representing multiple types of concept. Tag clouds for Wikipedia articles could be made up of associated articles and the main categories of the article. Similarly it was noted that MySpace members could be visualized through faceted tag clouds enabling the exploration by favorite music genres, main keywords in biographies, and what kind of artists people selected as friends. Another sketch featured a visual exploration interface for a library catalogue that would provide genre tag clouds. The person that drew this sketch explained it as a digital counterpart to browsing library shelves, which people should also be able to do online.

6.3.3 Possible Uses

Conversation during the focus group involved possible cases where visual information exploration could be useful or not. Participants did not see much need for visualizations in forums or webmail interfaces, as they considered the temporal organization of email and the topical folder structure in forums to be most useful. It was argued that in these cases visualization would add unnecessary overhead.

However, participants discussed multiple other possible applications and collections, such as music libraries, files and folders, Wikipedia articles, auctions, real estate offers, and photos. In these cases participants considered search and exploration with interactive visualizations a possibly useful alternative. Participants suspected that browsing through many information items about virtually anything—such as news, files, or auction items—requires multiple ways to search through them. Visual information exploration was seen as potentially helpful, especially for those information collections that are diverse and can be organized by several dimensions or aspects.

Comparing the VIEW approach with the general use of Google for any search, even for sites like YouTube or Wikipedia, one participant noted the universality of visual information exploration in that it could be applied to almost any kind of information.

6.3.4 Future Directions

As mentioned, the focus group should be seen rather as an informal exchange between the study experimenter and a few interested study participants. From the focus group no generalizable knowledge can be derived. However, some of the points raised during the conversation suggest interesting avenues for further research:

- The usefulness of visual information exploration may partly depend on the size and diversity of the information space to be explored.
- Use cases for different domains such as social networks, encyclopedic articles, and library catalogues may provide realistic applications and interesting use cases for visual information exploration.
- The three implemented VisGets may provide general building blocks that are applicable to many specific domains, whereas the attribution of meaning to each dimension would depend on the context.
- It is interesting to investigate to what degree VisGets may complement conventional Web search.

6.4 Summary

This chapter discussed how open-ended, exploratory evaluation has been undertaken to assess idea and realization of the VIEW concept. The research interest was focused on getting feedback from information seekers on the usefulness of visual information exploration, learning about problems and suggestions for improvement, and generating new ideas. A user study with 10 participants was conducted, that included different types of information seeking and exploration tasks. By means of questionnaires and semi-structured interviews, study participants were asked about their perception, opinion, and suggestions concerning the VisGets. The results have shown that VisGet were perceived to be useful for making information discoveries and uncovering relationships between information items. In particular the temporal VisGet was noted to be helpful. Suggestions for improvement mostly focused on the limitations of the implemented artifact.

To learn more about how information seekers think about the concept of visual information exploration, its usefulness, and to generate new ideas, an informal focus group was undertaken with four interested participants of the user study. Discussion ranged from present Web use, applications for visual information exploration to its limitations and potential for certain contexts. Participants noted that the VIEW approach would be especially helpful for information domains that are organized by multiple dimensions, such as news, files, and auction items. Whereas visualization was considered to add unnecessary overhead in cases where, for example, temporal order or folder structures are sufficient for finding and organizing information. During the discussion participants were asked to draw sketches of a useful application for visual information exploration. Afterwards the sketches were discussed in the focus group revealing that participants easily applied the general dimensions time, location, and tags to different, specific contexts and extended the scope of the initial VisGets. For example, two participants drew a tag VisGet as a faceted tag cloud of multiple types of concepts that they found to be possibly meaningful for sites such as MySpace, Wikipedia, and online library catalogues. The findings of the focus group cannot be generalized, but show interesting directions of future research.

The evaluation undertaken as part of this thesis project is an initial attempt to study how VisGets may support information seeking on the Web. The insights suggest that visual information exploration is a viable approach to improve navigation and search on the Web. However, more thorough studies and further use cases are needed to better understand the potential role of interactive visualizations in information seeking activity.

7 Conclusion

While the Web evolved into a universal information space and viable software platform for visualization, information seeking on the Web has remained rather low-level and laborious. Visualization has largely neglected to embrace the Web as an information space, so that information retrieval on the Web remained relatively unaffected by visualization. In this thesis I addressed the shortcomings of current information seeking and developed ideas for utilizing interactive visualizations with the intention of improving the way we find and explore information on the Web. This chapter summarizes the contributions presented in this work, indicates their limitations, and outlines possible directions for future work in the context of visual information exploration on the Web.

7.1 Summary

The main contributions of this thesis are organized into four categories: concept, design, implementation, and evaluation. In the following, I briefly summarize the results achieved in each of those.

Concept

In this work, I have developed the concept of visual information exploration on the Web (VIEW) with which I have proposed an interactive information visualization method with the goal of making information seeking a more engaging and fluid activity. InfoVis widgets, named VisGets, provide an interactive and visual way to explore diverse and distributed information resources on the Web. Several aspects of the information space are visualized by means of conceptual dimensions. These conceptual dimensions were chosen with the intention to use common topics which may prove meaningful to the information seeker. The searcher is envisioned as an active information seeker who is supported by a sophisticated, user-focused information system. Derived from the VIEW concept I proposed design goals for VisGets.

Design

VisGets combine visual overviews and navigation with interactive visualization widgets conceived to support the VIEW concept. I have designed three initial VisGets that provide functionality for visual information exploration along the temporal, spatial, and topical dimensions. VisGets are linked together to allow multi-dimensional query formulation and coordinated interaction across dimensions. They are embedded in a VIEW interface that is accessible using a Web browser. Changing query parameters by means of VisGets results in changes in the whole VIEW interface.

Implementation

I have implemented VisGets and the VIEW system in a Web-based environment, where interaction and visualization logic is running in the Web browser and data processing is carried out on the Web server. The VIEW system aggregates large amounts of distributed information items represented as RSS feeds and makes these available through VisGets running in the Web browser. To reduce data exchanged between client and server I developed Delta Queries that can be understood as a differential query mechanism.

Evaluation

The realized VIEW system and its approach were subjected to different exploratory evaluation methods with the goal to assess acceptance, find new applications, and generate further ideas. An initial user study undertaken with ten participants indicated positive reactions in particular towards the time VisGet. A follow-up focus group provided insight about the usefulness of VisGets and revealed interesting ideas for use cases of visual information exploration.

7.2 Limitations

The work presented in this thesis constitutes an early attempt to improve Web-based information seeking by the means of interactive visualizations. In the following I discuss limitations of the design, implementation, and evaluation of the VIEW approach.

Simple VisGets

The visual representations and query formulation realized by the VisGets are limited in their sophistication, which is partly due to constraints of the platform and the preliminary state of the initial design.

- A time VisGet only allows for intervals of months or days of one month. It is not possible to select a range of days across multiple months or to select certain temporal patterns, such as seasons or week days.

- The location VisGet assumes that an information source may only have one location associated with it. This, however, does not fit well with many information spaces, for example, news which often include multiple locations.
- The tag VisGet assumes keywords and does not work with the words used in the information source. However, it would be useful to be able to switch between those two modes. The way tag selections are currently combined to a logical conjunction does not allow more sophisticated Boolean queries consisting of negations or disjunctions.

While the information seeker can change query parameters by means of direct manipulation, there is little control provided over the way visual representations are displayed.

Scalability Constraints

In the implemented system, the whole set of information items that are part of a selection needs to be retrieved from the Web server to generate the visualizations of the VisGets within the Web browser. In case of large information spaces, this implies possibly large volumes of data to be transferred. Moving some of the visualization logic to the server may decrease the amount of data exchanged.

Initial Evaluation

Evaluation undertaken as part of this thesis can be considered a first attempt of studying how information seekers accept and react to visual information exploration. Due to the preliminary state of the approach there was no comparative study undertaken with major information seeking systems, such as Web search engines.

7.3 Future Work

To further the knowledge about visual information exploration as an information seeking approach, several phases of the research life cycle should be revisited and integrated into a conceptual framework. In addition to addressing the aforementioned limitations, particular problems that need to be solved by such a framework are briefly discussed in the following.

Formalize Design Process

The first step in developing a VIEW system is designing VisGets that are based on the characteristics of the information sources to be explored. Formalizing the design process may facilitate the reuse of previously created VisGets and the creation of novel VisGets for information spaces with other dimensions than time, location, and tags. It is essential to carefully analyze a given information space and clearly describe its conceptual dimensions for developing visual representations and query mechanisms that are meaningful and understandable for information seekers. Experiences gained through use cases with different information spaces may be integrated into a taxonomy that ties in dimensionality of information with types of visualization and interactivity.

Ease Implementation

Realizing visual information exploration is a difficult undertaking as it involves multiple programming languages and development contexts spread across the visualization pipeline. To ease the implementation of VIEW systems, common solutions for development challenges such as Delta Queries (see Section 5.3) should be integrated into a coherent form. These experiences can be part of a pattern library and a software toolkit. For example, commonly used functionality could be unified in a VIEW software library that would allow developers and researchers to innovate on the basis of previous achievements. The implementation of VisGets within existing information spaces requires knowledge about data structures and software infrastructures. Methods for aggregation, analysis, and integration of common data sources should be integrated into such a VIEW toolkit to provide well-defined interfaces.

Conduct Further Evaluation and Case Studies

In addition to evaluating viability and usefulness of visual information exploration, further research should give answers to the following questions:

- How do information seekers interact with VIEW interfaces?
- Which modes of information seeking are supported or neglected by VisGets?
- How do information seekers switch between different information seeking modes when using a VIEW system?

Learning about how people engage in visual information exploration requires extensive user research and diverse evaluation methods. Comparing VIEW interfaces with conventional search systems may uncover strengths and weaknesses of VisGets. Case studies for specific information spaces such as library catalogues and the Wikipedia may provide valuable feedback from information seekers with 'real' information needs.

Bibliography

[1] ACM SIGCHI. Curricula for human-computer interaction. `http://sigchi.org/cdg/cdg2.html` (Retrieved 2008-05-14).

[2] B. Adida and M. Birbeck. RDFa primer: Embedding structured data in web pages. Working draft, World Wide Web Consortium (W3C), March 17, 2008.

[3] S. Ahern, M. Naaman, R. Nair, and J. H.-I. Yang. World explorer: Visualizing aggregate data from unstructured text in geo-referenced collections. In *JCDL '07: Proceedings of the 7th ACM/IEEE joint conference on Digital libraries*, pages 1–10. ACM Press, 2007.

[4] C. Ahlberg and B. Shneiderman. Visual information seeking: Tight coupling of dynamic query filters with starfield displays. In *CHI '94: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 313–317. ACM Press, 1994.

[5] J.-W. Ahn, P. Brusilovsky, D. He, J. Grady, and Q. Li. Personalized web exploration with task models. In *WWW '08: Proceedings of the 17th international conference on World Wide Web*, 2008.

[6] N. Alper and C. Stein. Geospatial metadata querying and visualization on the WWW using Java applets. In *INFOVIS 1996. IEEE Symposium on Information Visualization*, pages 77–84. IEEE Computer Society, 1996.

[7] K. Andrews. Evaluating information visualisations. In *BELIV '06: Proceedings of the 2006 AVI workshop on BEyond time and errors*, pages 1–5. ACM Press, 2006.

[8] A. Ankolekar, M. Kroetzsch, T. Tran, and D. Vrandecic. The two cultures: Mashing up web 2.0 and the semantic web (position paper). In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, 2007.

[9] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives. DBpedia: A nucleus for a web of open data. In *International Semantic Web Conference*, 2007.

[10] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison Wesley, May 1999.

[11] M. Q. W. Baldonado, A. Woodruff, and A. Kuchinsky. Guidelines for using multiple views in information visualization. In *AVI '00: Proceedings of the working conference on Advanced visual interfaces*, pages 110–119. ACM Press, 2000.

[12] L. Barkhuus and J. A. Rode. From mice to men-24 years of evaluation in CHI. In *CHI '07 extended abstracts on Human factors in computing systems*. ACM Press, 2007.

[13] M. J. Bates. The design of browsing and berrypicking techniques for the online search interface. *Online Information Review*, 13(5):407–424, 1989.

[14] N. J. Belkin. Interaction with texts: Information retrieval as information-seeking behavior. *Information Retrieval*, 93:55–66, 1993.

[15] T. Berners-Lee. The website of the world's first-ever web server. http://info.cern.ch/ (Retrieved 2008-05-14), 1990.

[16] T. Berners-Lee, J. Hendler, and O. Lassila. The semantic web. *Scientific American*, 284(5):28–37, 2001.

[17] N. Boukhelifa and P. J. Rodgers. A model and software system for coordinated and multiple views in exploratory visualization. *Information Visualization*, 2(4):258–269, 2003.

[18] A. Buja, J. A. McDonald, J. Michalak, and W. Stuetzle. Interactive data visualization using focusing and linking. In *VIS '91: Proceedings of the 2nd conference on Visualization*, pages 156–163. IEEE Computer Society, 1991.

[19] V. Bush. As we may think. *The Atlantic Monthly*, July 1945.

[20] CakePHP. http://cakephp.org/ (Retrieved 2008-05-14).

[21] S. K. Card, J. D. Mackinlay, and B. Shneiderman. *Using vision to think*, pages 579–581. Morgan Kaufmann Publishers Inc. San Francisco, CA, USA, 1999.

[22] S. K. Card, G. G. Robertson, and J. D. Mackinlay. The information visualizer, an information workspace. In *CHI '91: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 181–186. ACM Press, 1991.

[23] S. K. Card, G. G. Robertson, and W. York. The webbook and the web forager: an information workspace for the world-wide web. In *CHI '96: Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM Press, 1996.

[24] E. H. Chi, P. Pirolli, K. Chen, and J. Pitkow. Using information scent to model user information needs and actions and the Web. In *CHI '01: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 490–497. ACM Press, 2001.

[25] C. Collins and S. Carpendale. VisLink: Revealing relationships amongst visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 13(6), Nov-Dec 2007.

[26] Del.icio.us. http://del.icio.us (Retrieved 2008-05-14).

[27] G. Ellis and A. Dix. An explorative analysis of user evaluation studies in information visualisation. In *BELIV '06: Proceedings of the 2006 AVI workshop on BEyond time and errors*, pages 1–7. ACM Press, 2006.

[28] J. English, M. Hearst, R. Sinha, K. Swearingen, and K.-P. Yee. Flexible search and navigation using faceted metadata. In *25th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2002)*, pages 11–15, August 2002.

[29] J. English, M. A. Hearst, R. Sinha, K. Swearingen, and K.-P. Yee. Hierarchical faceted metadata in site search interfaces. In *CHI '02 extended abstracts on Human factors in computing systems*, pages 628–639. ACM Press, 2002.

[30] Facebook. http://facebook.com (Retrieved 2008-05-14).

[31] G. Fischer. User modeling in human–computer interaction. *User Modeling and User-Adapted Interaction*, 11(1):65–86, 2001.

[32] P. M. Fitts. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology*, 47(6):262–269, June 1954.

[33] Flickr. http://flickr.com (Retrieved 2008-05-14).

[34] Flickr Map. http://flickr.com/map (Retrieved 2008-05-14).

[35] J. J. Garrett. Ajax: A new approach to web applications. http://www.adaptivepath.com/publications/essays/archives/000385.php (Retrieved 2008-05-14), February 2005.

[36] GeoNames. http://www.geonames.org/ (Retrieved 2008-05-14).

[37] Global Voices Online. http://www.globalvoicesonline.org/ (Retrieved 2008-05-14).

[38] S. Golder and B. A. Huberman. Usage patterns of collaborative tagging systems. *Journal of Information Science*, 32(2):198–208, April 2006.

[39] E. Golub and B. Shneiderman. Dynamic query visualisations on world wide web clients: A DHTML solution for maps and scattergrams. *International Journal of Web Engineering and Technology*, 1(1):63–78, 2003.

[40] Google. http://www.google.com (Retrieved 2008-05-14).

[41] Google Maps. http://maps.google.com (Retrieved 2008-05-14).

[42] S. Greenberg and B. Buxton. Usability evaluation considered harmful (some of the time). In *CHI '08: Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM Press, 2008.

[43] J. Harris and S. Kamvar. We Feel Fine. http://www.wefeelfine.org/ (Retrieved 2008-05-14), 2006. http://www.wefeelfine.org/.

[44] Y. Hassan-Montero and V. Herrero-Solana. Improving tag-clouds as visual information retrieval interfaces. In *Proceedings of the International Conference on Multidisciplinary Information Sciences and Technologies, InSciT2006.*, 2006.

[45] C. G. Healey, K. S. Booth, and J. T. Enns. High-speed visual estimation using preattentive processing. *ACM Transactions on Computer-Human Interaction*, 3(2):107–135, 1996.

[46] M. A. Hearst. Tilebars: Visualization of term distribution information in full text information access. In *CHI '95: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 59–66. ACM Press/Addison-Wesley, 1995.

[47] M. A. Hearst and C. Karadi. Cat-a-cone: an interactive interface for specifying searches and viewing retrieval results using a large category hierarchy. In *SIGIR '97: Proceedings of the 20th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 246–255. ACM Press, 1997.

[48] R. Hendley, N. Drew, A. Wood, and R. Beale. Case study: Narcissus: visualising information. In *INFOVIS 1995. IEEE Symposium on Information Visualization*. IEEE Computer Society, 1995.

[49] W. E. Hick. On the rate of gain of information. *Quarterly Journal of Experimental Psychology*, 4:11–26, 1952.

[50] D. F. Huynh, D. R. Karger, and R. C. Miller. Exhibit: Lightweight structured data publishing. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 737–746. ACM Press, 2007.

[51] B. Johnson and B. Shneiderman. Tree-maps: a space-filling approach to the visualization of hierarchical information structures. In *VIS '91: Proceedings of the 2nd conference on Visualization*, pages 284–291. IEEE Computer Society, 1991.

[52] S. Jones. VQuery: A graphical user interface for boolean query specification and dynamic result preview. *Annual Symposium on User Interface Software and Technology*, pages 143–151, November 1999.

[53] jQuery. `http://jquery.com/` (Retrieved 2008-05-14).

[54] O. Kaser and D. Lemire. Tag-cloud drawing: Algorithms for cloud visualization. In *Tagging and Metadata for Social Information Organization at WWW '07*, 2007.

[55] A. Kay and A. Goldberg. Personal dynamic media. *IEEE Computer*, 10(3):31–41, 1977.

[56] P. Klein, F. Muller, H. Reiterer, and M. Eibl. Visual information retrieval with the supertable + scatterplot. In *Information Visualisation, 2002. Proceedings. Sixth International Conference on*, pages 70–75, 2002.

[57] S. Koshman. Exploratory search visualization: Identifying factors affecting evaluation. In *Evaluating Exploratory Search Systems (Workshop)*, 2006.

[58] W. Kraaij and W. Post. Task based evaluation of exploratory search systems. In *Evaluating Exploratory Search Systems (Workshop)*, 2006.

[59] C. Kuhlthau. Inside the search process: Information seeking from the user's perspective. *Journal of the American Society for Information Science*, 42(5):361–371, 1991.

[60] B. Y.-L. Kuo, T. Hentrich, B. M. . Good, and M. D. Wilkinson. Tag clouds for summarizing web search results. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 1203–1204. ACM Press, 2007.

[61] T. Landauer. *The Trouble with Computers: Usefulness, Usability, and Productivity*, chapter Usefulness and Usability, pages 141–168. MIT Press, 1995.

[62] J. Licklider. Man-computer symbiosis. *IRE Transactions on Human Factors in Electronics*, HFE-1:4–11, March 1960.

[63] J. D. Mackinlay. Automating the design of graphical presentations of relational information. *ACM Transactions on Graphics*, 5(2):110–141, 1986.

[64] J. D. Mackinlay, G. G. Robertson, and S. K. Card. The perspective wall: detail and context smoothly integrated. In *CHI '91: Proceedings of the SIGCHI conference on Human factors in computing systems*, volume 27, pages 173–176, 1991.

[65] MagpieRSS. `http://magpierss.sourceforge.net/` (Retrieved 2008-05-14).

[66] T. M. Mann and H. Reiterer. Case study: A combined visualization approach for WWW-search results. In *INFOVIS 1999. IEEE Symposium on Information Visualization*. IEEE Computer Society, 1999.

[67] G. Marchionini. Exploratory search: from finding to understanding. *Communications of the ACM*, 49(4):41–46, 2006.

[68] MarkMail. `http://markmail.org/search/?q` (Retrieved 2008-05-07).

[69] R. Mazza and A. Berre. Focus group methodology for evaluating information visualization techniques and tools. *Information Visualization, 2007. IV '07. 11th International Conference*, pages 74–80, 4-6 July 2007.

[70] Microformats. `http://microformats.org/` (Retrieved 2008-05-14).

[71] A. Mockus, S. Hibino, and T. Graves. A web-based approach to interactive visualization in context. In *AVI '00: Proceedings of the working conference on Advanced visual interfaces*, pages 181–188. ACM Press, 2000.

[72] T. Munzner and P. Burchard. Visualizing the structure of the world wide web in 3d hyperbolic space. *Proceedings of the first symposium on Virtual reality modeling language*, pages 33–38, 1995.

[73] B. Myers. A brief history of human-computer interaction technology. *interactions*, 5(2):44–54, 1998.

[74] T. H. Nelson. Getting it out of our system. In G. Schlechter, editor, *Information Retrieval: A Critical Review*. Thompson Books, Washington, D.C., 1967.

[75] C. North. Toward measuring visualization insight. *IEEE Computer Graphics and Applications*, 26(3):6–9, 2006.

[76] C. North and B. Shneiderman. Snap-together visualization: Can users construct and operate coordinated visualizations? *International Journal of Human-Computers Studies*, 53(5):715–739, 2000.

[77] L. T. Nowell, R. K. France, D. Hix, L. S. Heath, and E. A. Fox. Visualizing search results: some alternatives to query-document similarity. In *SIGIR '96: Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 67–75. ACM Press, 1996.

[78] T. O'Reilly. What is web 2.0 – design patterns and business models for the next generation of software. `http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html` (Retrieved 2008-05-14), November 2005.

[79] P. Pirolli and S. K. Card. Information foraging in information access environments. In *CHI '95: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 51–58. ACM Press, 1995.

[80] C. Plaisant. The challenge of information visualization evaluation. In *AVI '04: Proceedings of the working conference on Advanced visual interfaces*, pages 109–116. ACM Press, 2004.

[81] C. Plaisant, B. Shneiderman, K. Doan, and T. Bruns. Interface and data architecture for query preview in networked information systems. *ACM Transactions on Information Systems*, 17(3):320–341, 1999.

[82] D. Raggett, A. L. Hors, and I. Jacobs. HTML 4.01 specification. `http://www.w3.org/TR/html4/` (Retrieved 2008-05-14), December 1999.

[83] L. Reeve, H. Han, and C. Chen. Information visualization and the semantic web. In V. Geroimenko and C. Chen, editors, *Visualizing the Semantic Web: XML-Based Internet and Information Visualization*. Springer, 2006.

[84] M. Rester and M. Pohl. Methods for the evaluation of an interactive infovis tool supporting exploratory reasoning processes. In *BELIV '06: Proceedings of the 2006 AVI workshop on BEyond time and errors*, pages 1–6. ACM Press, 2006.

[85] A. W. Rivadeneira, D. M. Gruen, M. J. Muller, and D. R. Millen. Getting our head in the clouds: Toward evaluation studies of tagclouds. In *CHI '07: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 995–998. ACM Press, 2007.

[86] G. G. Robertson, J. D. Mackinlay, and S. K. Card. Cone trees: Animated 3d visualizations of hierarchical information. In *CHI '91: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 189–194. ACM Press, 1991.

[87] R. M. Rohrer and E. Swing. Web-based information visualization. *IEEE Computer Graphics and Applications*, 17(4):52–59, 1997.

[88] A. Scharl and K. Tochtermann, editors. *The Geospatial Web – How Geobrowsers, Social Software and the Web 2.0 are Shaping the Network Society*. Springer, 2007.

[89] W. Schumann, Heidrun und Müller. *Visualisierung, Grundlagen und allgemeine Methoden*. Springer, 2000.

[90] B. Shneiderman. Dynamic queries for visual information seeking. *IEEE Software*, 11(6):70–77, 1994.

[91] B. Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *Proceedings of the IEEE Symposium on Visual Languages*, pages 336–343, 1996.

[92] A. Slingsby, J. Dykes, J. Wood, and K. Clarke. Interactive tag maps and tag clouds for the multiscale exploration of large spatio-temporal datasets. In *IV '07: Proceeding of the International Conference Information Visualization*, pages 497–504, 2007.

[93] R. Spence and L. Tweedie. The attribute explorer: information synthesis via exploration. *Interacting with Computers*, 11(2):137–146, 1998.

[94] A. Spoerri. Infocrystal: A visual tool for information retrieval. In *VIS '93: Proceedings of the 4th conference on Visualization*, pages 150–157, 1993.

[95] E. Stoica, M. Hearst, and M. Richardson. Automating creation of hierarchical faceted metadata structures. In *Proceedings of NAACL HLT 2007*, pages 244–251. Association for Computational Linguistics, April 2007.

[96] I. Sutherland. Sketchpad, a man-machine communication system. *Proceedings of AFIPS Spring Joint Computer Conference*, 23:329–346, 1963.

[97] Technorati. `http://technorati.com/` (Retrieved 2008-05-14).

[98] K. Toyama, R. Logan, and A. Roseway. Geographic location tags on digital images. In *MULTIMEDIA '03: Proceedings of the 11th ACM international conference on Multimedia*, pages 156–166. ACM Press, 2003.

[99] C. J. Van Rijsbergen. *Information Retrieval, 2nd edition*. Dept. of Computer Science, University of Glasgow, 1979.

[100] A. Veerasamy and N. J. Belkin. Evaluation of a tool for visualization of information retrieval results. In *SIGIR '96: Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 85–92. ACM Press, 1996.

[101] F. Viégas, M. Wattenberg, F. van Ham, J. Kriss, and M. McKeon. Many eyes: A site for visualization at internet scale. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1121–1128, Nov/Dec 2007.

[102] M. Wattenberg. Baby names, visualization, and social data analysis. In *INFOVIS 2005. IEEE Symposium on Information Visualization*, pages 1–7. IEEE Computer Society, October 2005.

[103] R. W. White, B. Kules, S. M. Drucker, and m.c. schraefel. Supporting exploratory search – introduction. *Communications of the ACM*, 49(4):36–39, 2006.

[104] M. Wick and T. Becker. Enhancing RSS feeds with extracted geospatial information for further processing and visualization. In Scharl and Tochtermann [88], pages 105–116.

[105] Wikipedia. `http://en.wikipedia.org` (Retrieved 2008-05-14).

[106] Wikipedia. Precision and recall. `http://en.wikipedia.org/w/index.php?title=Precision_and_recall&oldid=203980071` (Retrieved 2008-05-14), 2008.

[107] W. Willett, J. Heer, and M. Agrawala. Scented widgets: Improving navigation cues with embedded visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1129–1136, Nov-Dec 2007.

[108] T. Wilson. Human Information Behavior. *Informing Science The International Journal of an Emerging Transdiscipline*, 3(2):49–56, 2000.

[109] J. Wood, K. Brodlie, and H. Wright. Visualization over the world wide web and its application to environmental data. In *VIS '96: Proceedings of the 7th conference on Visualization*, page 81, Los Alamitos, CA, USA, 1996. IEEE Computer Society.

[110] J. Wood, J. Dykes, A. Slingsby, and K. Clarke. Interactive visual exploration of a large spatio-temporal dataset: Reflections on a geovisualization mashup. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1176–1183, November 2007.

[111] Yahoo! Search. MediaRSS specification. `http://search.yahoo.com/mrss` (Retrieved 2008-05-14).

[112] K.-P. Yee, K. Swearingen, K. Li, and M. A. Hearst. Faceted metadata for image search and browsing. In *CHI '03: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 401–408. ACM Press, 2003.

Independence Statement

Herewith I declare that I have completed this work solely and with only the help of the mentioned references.

Marian Dörk
Magdeburg, May 19, 2008